



Sistemas Informáticos

Curso 2003-04

Diseño de un interfaz gráfico para procesamiento avanzado de imágenes

M^a Luisa Carles Gil
Javier Espejo García
Cristina García Aguirre

Dirigido por:
Prof. Gonzalo Pajares Martisanz
Dpto. de Arquitectura de Computadores y
Automática

Facultad de Informática
Universidad Complutense de
Madrid

Facultad De Informática
U.C.M



Departamento de Arquitectura de Computadores y
Automática

Sistemas Informáticos
Curso 2003-2004



*Diseño de un interfaz gráfico para procesamiento avanzado
de imágenes*

EDIT PHOTO PRO v.1.0

ÍNDICE DE CONTENIDOS

1. GLOSARIO DE TÉRMINOS.....	1
2. INTRODUCCIÓN.....	3
DESCRIPCIÓN DEL PROYECTO.....	3
PROJECT DESCRIPTION.....	4
ANTECEDENTES.....	5
3. VIABILIDAD.....	6
4. ESPECIFICACIÓN DE REQUISITOS.....	7
5. ANÁLISIS.....	7
6. ¿QUÉ ES UNA IMAGEN DIGITAL?	7
7. FUNCIONES DE ÁLBUM	9
8. FUNCIONES DE ACTUALIZACIÓN EN INTERNET	9
9. HERRAMIENTAS DE PROGRAMACIÓN.....	10
LENGUAJE JAVA. DESCRIPCIÓN Y CUALIDADES	10
LAS BIBLIOTECAS JAI DE SUN MICROSYSTEM	16
10. DISEÑO DE LA ARQUITECTURA.....	16
MODULARIDAD	19
REUSABILIDAD	20
11. DISEÑO DETALLADO.....	20
JAI EN NUESTRA APLICACIÓN	21
SENCILLEZ DE MANEJO	22
12. PRUEBAS DE SISTEMA.....	24
13. IMPLANTACIÓN Y CIERRE DEL PROYECTO.....	24
14. TRABAJO FUTURO.....	24
PROCESAMIENTO DE IMÁGENES AVANZADO	24

AMPLIACIÓN DE FUNCIONALIDADES	28
15. BIBLIOGRAFÍA.....	29
16. ANEXOS.....	30
ANEXO A: Estándar de Documentación	31
ANEXO B: Gestión de Configuración	33
ANEXO C: Análisis de riesgos	36
ANEXO D: Especificación de requisitos	42
ANEXO E: Análisis del sistema	45
ANEXO F: Cómo añadir elementos a la interfaz	49
ANEXO G: Cómo añadir una nueva transformación ...	52
ANEXO H: Instalación	55
ANEXO I: Manual de usuario	57

1. GLOSARIO DE TÉRMINOS

- Álbum: colección de fotos agrupadas en una plantilla html.
- Banda: cada una de las matrices de valores que conforman una imagen. La composición de los valores de las diferentes matrices para una misma fila y columna da como resultado un píxel. También se le llama *canal*.
- Brillo: luminosidad del color negro en un monitor.
- Capa: cada una de las imágenes atómicas de que se compone una imagen en la aplicación.
- Contraste: factor de ganancia que se aplica a las señales roja, verde y azul de una imagen.
- Evento: mecanismo por el que un objeto notifica de manera indirecta su estado al resto de los objetos.
- Extracción de color: procedimiento según el cual se transforma una imagen a blanco y negro aplicando a cada banda una transformación umbral (o su inversión) y computando la AND lógica de las bandas.
- Filtro: transformación de imagen que elimina información.
- Formato BMP: acrónimo de Bit Map, formato de almacenamiento de gráficos sin compresión de información.
- Formato GIF: acrónimo de CompuServes Graphics Interchange Format, formato de compresión de archivos gráficos sin pérdida de información limitado a gráficos con un máximo de 256 colores.
- Formato JPEG: acrónimo de Joint Photographic Experts Group, formato de compresión de archivos gráficos con pérdida de información.
- Formato PNG: acrónimo de Portable Network Graphics, formato de compresión de archivos gráficos sin pérdida de información.
- Formato PNM: acrónimo de Portable AnyMap, formato de compresión de archivos gráficos sin pérdida de información.
- Formato TIFF: acrónimo de Tagged Image File Format, formato de compresión de archivos gráficos sin pérdida de información.
- FTP: (File Transfer Protocol) Consiste en la transferencia de ficheros de un ordenador a otro en Internet. Generalmente se utiliza para el intercambio de productos informáticos y programas.
- GUI: acrónimo de Graphic User Interface, interfaz gráfica de usuario.
- Herramienta: utilidad genérica de gestión, observación o manipulación de imágenes.
- Histograma: representación gráfica del análisis de frecuencia de los valores de muestreo estadístico. En tratamiento de imágenes, los valores de muestreo son los valores de intensidad de color.
- HSI: acrónimo de Hue – Saturation – Intensity, formato de representación del color mediante su tono, saturación e intensidad.

- HTML: (*HyperText Markup Language*) es un lenguaje muy sencillo que permite describir hipertexto, es decir, texto presentado de forma estructurada y agradable, con *enlaces* (*hyperlinks*) que conducen a otros documentos o fuentes de información relacionadas, y con *inserciones* multimedia (gráficos, sonido...) La descripción se basa en especificar en el texto la estructura lógica del contenido (títulos, párrafos de texto normal, enumeraciones, definiciones, citas, etc) así como los diferentes efectos que se quieren dar (especificar los lugares del documento donde se debe poner cursiva, negrita, o un gráfico determinado) y dejar que luego la presentación final de dicho hipertexto se realice por un programa especializado (como Mosaic, o Netscape).
- Intensidad: luminosidad de un color.
- Internet: Algunas veces llamado simplemente "La Red", es un sistema mundial de redes de computadoras, un conjunto integrado por las diferentes redes de cada país del mundo, por medio del cual un usuario en cualquier computadora puede, en caso de contar con los permisos apropiados, acceder información de otra computadora y poder tener inclusive comunicación directa con otros usuarios en otras computadoras.
- JAI: acrónimo de Java Advanced Imaging, librería de componentes Java sobre tratamiento avanzado de imágenes.
- JFC: acrónimo de Java Foundation Classes, extensión de Java que proporciona componentes para la construcción de interfaces gráficas de usuario (GUI).
- Píxel: contracción de picture element, unidad básica de color programable de una imagen digitalizada.
- Plug-In: clase Java autogestionada diseñada para ser directamente acoplada a cualquier aplicación gracias a su inmediata interfaz.
- Portapapeles: almacén local para intercambio de información.
- Reflexión: transformación especular de la imagen.
- RGB: acrónimo de Red – Green – Blue, formato de representación de color mediante la composición de luminosidades en rojo, verde y azul.
- Saturación: índice de pureza de un color representado por la proporción no gris del mismo.
- Sitio Web: Es un conjunto de archivos electrónicos y páginas Web referentes a un tema en particular, que incluye una página inicial de bienvenida, generalmente denominada home page, con un nombre de dominio y dirección en Internet específicos.
- Tono: representación de la tonalidad de color real.
- Transformación: manipulación de una imagen.

2. INTRODUCCIÓN

2.1 Descripción del proyecto

El proyecto consiste en el desarrollo de un sistema software de tratamiento de imágenes digitales. En concreto, tiene como objetivo crear una aplicación que permita a fotógrafos (aficionados o profesionales) la edición y postproducción fotográfica de imágenes (encuadres, contraste, brillo, corrección de sobreexposición, etc) de manera fácil e intuitiva.

El proyecto se encuentra estructurado en tres partes diferenciadas:

- A) Edición y tratamiento general de imágenes.
Esta primera fase del proyecto abarca el tratamiento básico de una imagen digital.
- B) Funciones de álbum.
En este apartado se añaden a la aplicación funcionalidades referentes a la posibilidad de crear álbumes de las imágenes tratadas.
- C) Funciones de actualización en Internet.
Posibilidad de crear un entorno en la red para albergar los resultados obtenidos de nuestras imágenes.

Para la implementación del proyecto se va a emplear el lenguaje de programación JAVA debido a que es un lenguaje multiplataforma que puede ser ejecutado sobre diferentes sistemas.

Para la implementación de las operaciones sobre las imágenes a bajo nivel se utilizarán las librerías JAI, a través de las cuales se consigue un acceso de lectura y escritura a las imágenes píxel por píxel. De esta manera se puede acceder al formato de color de una imagen y aplicando los diferentes algoritmos se puede implementar cualquier operación sobre dicha imagen.

Por último, se orientará el proyecto para que pueda servir como base para proyectos futuros de tratamiento de imágenes.

2.2 Project description

The Project consists in the elaboration of a software system, designed to treat all sorts of digital images. The objective is to create an application that permits the photo to be modified in a quick and easy way. Useful for the beginner and the professional, the software allows the photo to be changed in all of its stages (correction of hue, colour, saturation, etc...).

The project has been structured around three points:

- A) General treatment of photo.
The first phase permits basic changes to be made.
- B) Album.
An application is added to permit the user to create personalized albums.
- C) Uploading to Internet.
Possibility to upload the changed images to a site on the Internet.

Java (as it is a multiplatform language program) has been used throughout the whole project as it can be executed on all sorts of different systems.

Using JAI libraries images can be read or written over pixel by pixel. This technique permits access to the colour format of each images and using all the different possible methods, the images can be manipulated in every way.

Finally, the project has been orientated so that it can be used as a platform for all sorts of future programs.



2.3 Antecedentes

Los antecedentes directos de este proyecto, son los desarrollados en años anteriores, dentro del Departamento de Arquitectura de Computadores y Automática. Aunque el enfoque de nuestro proyecto no se centra en el ámbito de imágenes médicas, tiene en común el tratamiento realizado sobre imágenes digitales.

Igualmente existen otras aplicaciones en el mercado de pretensiones similares a las nuestras. A continuación, describimos las más relevantes:

- ADOBE PHOTOSHOP:

Photoshop es la mejor aplicación para la manipulación de imagen digital, ya sea para medios impresos, Internet o multimedia. Tradicionalmente, Photoshop ha sido considerado como un estándar dentro de la industria de la impresión. Es posible que una gran cantidad de imágenes fotográficas que hayan visto impresas (carteles, libros, revistas, flyers, anuncios espectaculares, etc.) hayan sido preparados en Photoshop.

Las características que han hecho de Photoshop un estándar en la industria de medios impresos, son también muy útiles para Internet y multimedia. El programa ha evolucionado y mejorado sus capacidades de impresión y agregando características especiales para los que nos dedicamos al desarrollo en Web. Junto con Photoshop, actualmente se distribuye Image Ready, programa enfocado al diseño Web.

La última entrega de Photoshop incluye, entre otras características, la capacidad de aplicar la misma acción a múltiples capas seleccionadas, así como un set de paletas controlable a través de menú. Otros extras son las mejoras en el navegador de imágenes, el soporte para píxeles no-cuadrados, personalización de atajos de teclado mejorada y soporte para JPEG 2000.

- PAINT SHOP PRO:

Un completo programa de creación y manipulación de imágenes. Paint Shop Pro multiplica el poder de edición de los usuarios, tanto profesionales como aficionados, gracias a una facilidad de uso sin rival, velocidad y funcionalidad.

Incluye potentes herramientas de edición, un soporte completo para la administración de ficheros, una interfaz sencilla y lineal, y otras características avanzadas que permiten crear y preparar imágenes para presentaciones de negocios, retoque fotográfico digital, páginas Web, y cualquier otra forma de comunicación visual.

Algunas de sus opciones son: soporte de múltiples capas, más de 40 tipos de ficheros gráficos soportados, compatibilidad con cámaras digitales, brochas de efectos que permiten crear sorprendentes efectos especiales, separaciones de color CMYK (canales de color Cyan, Magenta, Amarillo y Negro), opción de deshacer de varios niveles, soporte para tabletas digitalizadoras sensibles a la presión, capacidad de combinar imágenes con la opción "Aritmética de imágenes", posibilidad de crear tus propios filtros, soporte de administración de color independiente, compatibilidad con el formato de plug-ins de PhotoShop, creación de GIF animados, automatización de las tareas más pesadas, texto editable con formato de caracteres, herramientas de dibujo vectorial, rehacer multinivel, etc.

- COREL PHOTO PAINT:

Corel PHOTO-PAINT es una amplia aplicación de edición de imágenes que permite retocar y mejorar fotografías. Tanto si desea corregir el efecto de ojos rojos o problemas de exposición, recortar áreas de la imagen o crear y publicar imágenes en Web, Corel PHOTO-PAINT le proporciona potentes herramientas, que además de rápidas, son fáciles de usar.

3. VIABILIDAD

Muchas veces cuando se emprende el desarrollo de un proyecto los recursos y el tiempo no son realistas para su materialización. Sin embargo, tanto el plazo de realización como los recursos básicos necesarios, estaban marcados desde el comienzo del proyecto, a través de una planificación acordada con el tutor del proyecto, que permitiese un seguimiento de los avances, la detección de errores, la necesidad de nuevas funciones, así como una plataforma tecnológica que facilitara el trabajo a desarrollar.

Para ello, al principio establecimos:

- Usar el lenguaje de programación Java, en el entorno de Borland JBuilder v9.0
- Usar un estándar de documentación, que puede ser consultado en el Anexo A, "Estándar de Documentación"
- Usar un estándar de código, siguiendo las pautas marcadas por proyectos de años anteriores
- Un gestor de configuración, que puede ser consultado en el Anexo B, "Gestión de Configuración"

Por otra parte, la viabilidad y el análisis de riesgos están relacionados de muchas maneras, si el riesgo del proyecto es alto, la viabilidad de producir software de calidad se reduce. En el Anexo C, "Análisis de riesgos", se puede consultar la lista de riesgos.

Todos los anexos a los que hacemos referencia, han sido revisados durante la evolución del proyecto, para adecuarlos a los cambios que se han ido realizando, con objeto de favorecer el desarrollo del proyecto.

4. ESPECIFICACIÓN DE REQUISITOS

El sistema tratará imágenes en color (RGB e HSI), de intensidad de grises e imágenes binarias y soporta distintos formatos de almacenamiento (.jpg, .png, .tif, .bmp, etc.).

Se considera prioritario que la aplicación sea modular y fácilmente ampliable para que el trabajo que sea realizado en este proyecto pueda ser continuado en años sucesivos con procesamiento de imágenes más avanzado y más opciones de postproducción.

En el Anexo D, "Especificación de requisitos", se puede consultar la lista de requisitos.

5. ANÁLISIS

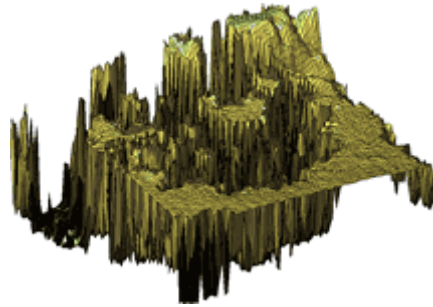
En esta fase se trata de modelar el sistema, es decir, se trata de decidir que debe hacer nuestra aplicación desde un punto de vista lógico.

Como la plataforma elegida es claramente orientada a objetos, para modelar el sistema, hemos usado el lenguaje UML.

En el Anexo E, "Análisis del sistema ", se observan los principales modelos desarrollados.

6. ¿QUÉ ES UNA IMAGEN DIGITAL?

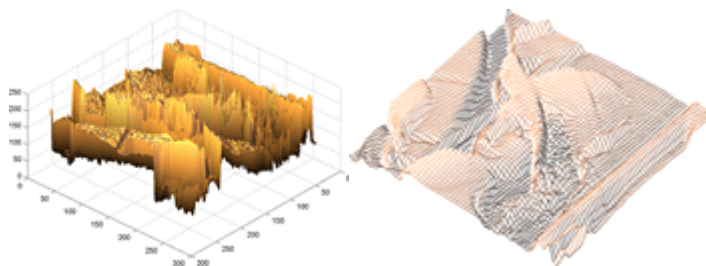
Las imágenes digitales son capturadas por dispositivos especiales, como cámaras digitales fotográficas o de vídeo, escáner u otros, analógicos o digitales, y son almacenadas como ficheros que, en el fondo, son una representación de una matriz rectangular, cada uno de cuyos elementos es el valor de píxel. Estamos hablando, en particular, de imágenes con niveles de gris, ya que una imagen en color se representa mediante una combinación lineal de las citadas imágenes.



Representación de la superficie de niveles de gris (0-255), con resolución reducida a 96x128.

Cuando se requiere la descripción digital de una imagen, es necesario decidir la cantidad de información que se atribuye a cada píxel. Esta cantidad de información se mide en número de bits por píxel. Para imágenes binarias, se hace uso de 2 bits, y la cantidad más habitual es 8 bits. La resolución depende del tamaño de la matriz (número total de píxeles, o número de filas y número de columnas de la matriz) y de la cantidad de bits que se asocia con cada píxel.

A continuación mostramos una imagen con escala de grises en un cuadrado de 256x256, de forma que cada uno de ellos lleva asociado un nivel de gris, representado por un número entero entre 0 y 255, caso en el que la cantidad de información asociada a cada píxel es de 8 bits.



Gráficas de niveles (256 y 64) de la imagen.

7. FUNCIONES DE ÁLBUM

Con la proliferación de las cámaras digitales es cada vez más común que de cada pequeño evento tengamos un número considerable de imágenes digitales. El usuario demanda un método sencillo, rápido y eficaz para poder ver sus imágenes de manera efectiva y organizada. La ausencia de estos métodos motiva la desazón que produce en muchas ocasiones la fotografía digital entre los usuarios, por considerarla “poco práctica”.

Por ello hemos implementado las funciones de álbum, que posibilitan a un usuario crear un muestrario de sus fotos de manera sencilla. A partir de una carpeta en la que ha ubicado una selección de fotos obtendrán un álbum con el que podrán organizar sus fotos con rapidez y comodidad.

Los álbumes son generados por la aplicación a partir de los parámetros definidos por el usuario (color de fondo, disposición de las fotografías, título...) y están en formato HTML.

Para posibilitar que los usuarios puedan utilizar una gran variedad de estilos y funcionalidades, la generación de los álbumes se realiza a partir de plantillas HTML que la aplicación parsea como un documento XML y adapta a los criterios definidos por el usuario.

Además con el uso del paquete org.w3c.dom reforzamos la universalidad de nuestra aplicación puesto que el estándar XML cada vez esta más difundido y permitiría fácilmente generar álbumes para dispositivos que puedan visionar documentos basados en XML como móviles, PDA's...

8. FUNCIONES DE ACTUALIZACIÓN EN INTERNET

Junto al desarrollo de los últimos años de la fotografía digital, la total y absoluta presencia de Internet en nuestra vida cotidiana hace de este medio el aliado perfecto de nuestras imágenes digitales. No cabe duda de que un objetivo primordial de nuestras fotografías es ser mostradas e Internet se ha convertido en el medio más rápido, cómodo y sencillo para hacerlo.

Por ello, la aplicación contiene una librería de funciones que permite al usuario colocar en Internet los álbumes que ha creado sin necesidad de utilizar aplicaciones externas y de manera transparente.

Las funciones de Internet están basadas en las librerías FtpBean v1.4.3 que nos proveen de funciones para manejar el protocolo FTP y de esta manera poder subir los álbumes al servidor web.

Esta funcionalidad junto a la creación de álbumes es la que dota a “Edit Photo Pro v1.0” de valor añadido que la convierten en una herramienta no solo de edición sino de publicación. De esta manera vemos cumplidos los objetivos de aunar en una sola aplicación las tareas básicas que un fotógrafo profesional o aficionado desea: visualización, edición, modificación y publicación de imágenes digitales de manera rápida, fácil e intuitiva.

Si el usuario no tiene espacio web la aplicación viene configurada con una cuenta gratuita. De esta manera posibilitamos el acercamiento del usuario no cualificado, que no desea adquirir otros programas ni realizar gestiones para conseguir espacio web, a la generación de contenidos en Internet.

9. HERRAMIENTAS DE PROGRAMACIÓN

9.1 Lenguaje Java. Descripción y cualidades

Los orígenes de Java se remontan al año 1990, en que un equipo de Sun Microsystems dirigido por James Gosling estaba desarrollando software para dispositivos electrónicos de consumo, es decir, para los chips de objetos tales como lavadoras o equipos de música. Estos microprocesadores deben programarse para determinadas funciones, pero los fabricantes cambian cada poco los chips a modelos más potentes y baratos, lo que obliga a reprogramarlos cada vez con nuevo software adaptado a ellos, que debe recompilarse. Se hizo necesario desarrollar un nuevo lenguaje de programación que permitiera escribir programas que funcionaran **en cualquier tipo de plataforma**, sin necesidad de efectuar modificaciones cada vez que se cambiara el modelo de chip (como había que hacer con lenguajes como C ó C++). Al mismo tiempo, el nuevo lenguaje debía ser capaz de crear programas pequeños y rápidos (pues este tipo de chips no tienen gran capacidad de proceso), a la par que robustos y fiables (un fallo en el software puede implicar la sustitución de un elemento importante del electrodoméstico). Así nació el lenguaje de programación Oak ("roble"), lanzado en Enero de 1991.

En 1993 se produce en **Internet** (hasta entonces limitada al ámbito académico) la creación de la denominada *World Wide Web* (www) y del primer navegador (Mosaic), que permite acercar dicha red a multitud de empresas y de usuarios particulares. Pronto se advierte la necesidad de instrumentos que permitan la interactividad de los usuarios con las páginas web, lo que propicia la aparición del estándar CGI (*Common Gateway Interface*): éste define los procedimientos utilizables para acceder desde una página web a un programa que reside y se ejecuta en el servidor web (en general, a través de un formulario). El estándar CGI tiene una utilidad limitada, pues implica un gran número de

conexiones que ralentizan las operaciones, además de sobrecargar el servidor. Así, en 1994, el equipo de Gosling, tras haber implementado Oak para un fallido proyecto de televisión interactiva, se percató de la potencialidad de su lenguaje para **mejorar la interactividad de las páginas web** en internet. De este modo, Oak se convierte en un nuevo lenguaje de programación, al que se llamará Java ("café hecho a la europea", en inglés de California), y el cual se lanza en mayo de 1995, junto con una versión preliminar de un navegador llamado HotJava ("café caliente"): con el nuevo lenguaje y el nuevo navegador, Sun aborda el problema de la interactividad desde un nuevo enfoque.

Este enfoque consiste en introducir como elementos de las páginas web los llamados **applets**. El navegador carga los applets junto con el resto de la página, y una vez cargados se ejecutan en el ordenador del usuario, visualizándose el resultado en la pantalla del navegador. La filosofía en la programación de los applets debe ser la misma que la que presidía la programación de los chips de los electrodomésticos: programas pequeños (para que se carguen en poco tiempo) y rápidos (para que se visualicen en la página web sin demoras), a la par que robustos y fiables (son inadmisibles errores o fallos del applet que pudieran modificar los ficheros del usuario o bloquear su sistema operativo); y sobre todo: han de funcionar en cualquier tipo de plataforma, con cualquier sistema operativo bajo el que funcione el navegador. El lenguaje y la plataforma Java permiten todo ello.

El hecho crucial para el éxito fulminante de Java fue la inclusión de la máquina virtual de Java en la versión 2.0 de **Netscape Navigator** (el navegador de Netscape Communications) bajo licencia de Sun, en septiembre de 1995. Poco después, deciden licenciar Java otras grandes compañías como IBM, Symantec o Microsoft (que integraría la máquina virtual en su navegador **Internet Explorer**).

Sun distribuye el primer entorno de programación en Java (Java Development Kit) en 1996, y entre 1996 y 1997 mejora drásticamente Java como lenguaje de programación, al incluir facilidades de impresión, el estándar JDBC de acceso a bases de datos, así como el estándar JavaBeans ("granos de café"): éste potencia el carácter de lenguaje orientado a objetos de Java, al tiempo que proporciona un entorno visual de programación que facilita y simplifica el desarrollo del software escrito en Java. De este modo, Java se convierte en un **poderoso entorno de programación y de ejecución de programas** para producir no sólo applets aislados para páginas web, sino también verdaderas aplicaciones de interés comercial, comparables a las programadas hasta ahora en otros lenguajes, pero con las ventajas mencionadas de independencia de plataforma (transportabilidad), robustez y fiabilidad de que gozaba el primitivo lenguaje Oak, así como de adaptación a la computación en red que ya tenía Java cuando irrumpió en Internet.

Cualidades principales de Java.

- **Universalidad.** Aunque hemos dicho que un programa interpretado no es en principio tan rápido como un programa equivalente compilado, las **prestaciones** de Java son sin embargo muchísimo mejores que las de cualquier lenguaje interpretado. Este hecho, junto con la **sencillez** de programación en Java (a la que nos referiremos luego) ha propiciado que se hayan escrito intérpretes de pequeño tamaño adaptados a prácticamente cualquier plataforma, desde mainframes y ordenadores personales (con cualquier sistema operativo: Windows, Macintosh OS, Unix, etc.) hasta dispositivos electrónicos de bajo coste. Además, la universalidad de los byte codes hacen de Java el lenguaje idóneo para desarrollar aplicaciones para Internet. De hecho, la mayor parte de los navegadores (Netscape Navigator, Internet Explorer, HotJava) integran máquinas virtuales, y por tanto intérpretes de Java, lo que hace posible acceder automáticamente a los applets presentes en las páginas web. De nuevo la sencillez de Java hace que esta integración no reduzca en absoluto las prestaciones de los navegadores, permitiendo además la **ejecución rápida y simultánea** de gran cantidad de applets.

También se suele hacer referencia a la universalidad de Java con términos equivalentes como **transportabilidad**, o **independencia de plataforma**, pues para ejecutar un programa basta compilarlo una sola vez: a partir de entonces, se puede hacer correr en cualquier máquina que tenga implementado un intérprete de Java.

Además, las bibliotecas estándar de funciones y métodos de Java (definidas en su API, *Application Programming Interface*) facilitan la programación de multitud de **acciones complejas** (desarrollo de **interfaces gráficas**, multimedia, **multitarea**, **interacción con bases de datos**,...). Ningún otro lenguaje (ni compilado ni interpretado) dispone como Java de una cantidad tan grande de **funciones accesibles en cualquier plataforma** sin necesidad de cambiar el código fuente.

- **Sencillez.** Java es un lenguaje de gran facilidad de aprendizaje, pues en su concepción se eliminaron todos aquellos elementos que no se consideraron absolutamente necesarios. Por ejemplo, en comparación con otros lenguajes como C ó C++, es notable la ausencia de punteros, o lo que es lo mismo: es imposible hacer referencia de forma explícita a una posición de memoria. Ello ahorra gran cantidad de tiempo a los programadores, dado que el comportamiento imprevisto de los punteros es una de las principales fuentes de errores en la ejecución de un programa. Por otra parte, el código escrito en Java es por lo general mucho más legible que el escrito en C ó C++.

Por otro lado, Java dispone de un mecanismo conocido como "recogida de basura", el cual (usando la capacidad multitarea de Java) hace que, durante la ejecución de un programa, los objetos que ya no se utilizan se eliminen automáticamente de la memoria. Dicho mecanismo facilita enormemente el diseño de un programa y optimiza los recursos de la máquina que se esté usando para la ejecución del mismo (con los lenguajes tradicionales, la eliminación de objetos y la consiguiente optimización de recursos debe planificarse cuidadosamente al idear el programa).

- **Orientación a objetos.** Aunque C++ es también, como Java, un lenguaje orientado a objetos, la diferencia fundamental es que Java lo es desde su concepción, mientras que C++ se diseñó como un lenguaje compatible con C (que no es orientado a objetos): de este modo, un programa escrito en C++ puede ignorar la mayoría de las características de orientación a objetos de C++ y compilarse sin problemas en un compilador de C++. Sin embargo, un programador no puede obviar la orientación a objetos cuando escribe un programa en Java, y esto hace que las aplicaciones escritas en Java tengan interesantes ventajas.

Para entender qué es la orientación a objetos, recordemos que los lenguajes tradicionales no orientados a objetos, como Pascal ó C, estaban pensados para trabajar de forma secuencial y basaban su funcionamiento en el concepto de procedimiento o función. La tarea principal del programa se dividía en funciones o tareas más pequeñas, a menudo muy interrelacionadas, por lo que era difícil modificar una función sin tener que revisar el resto del programa: por tanto, también era difícil reutilizar o actualizar los programas ya escritos.

En el caso de los lenguajes orientados a objetos, el concepto clave no es el de función, sino el de objeto. Un objeto es un elemento de programación, autocontenido y reutilizable, y que podríamos definir como la representación en un programa de un concepto, representación que está formada por un conjunto de variables (los datos) y un conjunto de métodos (o instrucciones para manejar los datos). Por ejemplo, en una aplicación de Bibliotecas un objeto puede ser un código de barras, que contiene datos (por ejemplo el número mismo del código de barras) e instrucciones para manejarlos (por ejemplo el método para calcular el dígito de control). Los objetos, además, poseen la capacidad de enviarse mensajes entre sí durante la ejecución de un programa.

La "encapsulación" de variables y métodos en un objeto tiene claras ventajas:

- Cada objeto puede ser modificado y mantenido por separado.
- Se pueden mantener en un objeto métodos y variables que no son accesibles desde fuera de él, lo que evita multitud de posibilidades de error en el momento de confeccionar un programa. Esta característica se llama "ocultamiento de la información".
- Es posible reutilizar porciones de programa ya escritas: en el ejemplo anterior, un programador puede reutilizar el objeto "código de barras" para escribir software para diversos tipos de transacciones, como préstamo, devolución, reserva, etc. sin necesidad de reescribir cada vez los comandos para calcular el dígito de control, dado que el programa se encargará de hacerlo cada vez que se use el objeto.

Como hemos visto, el estándar JavaBeans proporciona un entorno visual de programación que facilita y simplifica la programación de objetos. De hecho, es posible escribir applets sencillos en Java sin necesidad de escribir una sola línea de código.

- **Seguridad.** En general, se considera que un lenguaje es tanto más seguro cuanto menor es la posibilidad de que errores en la programación, o diseños malintencionados de programas (virus), causen daños en el sistema. La extrema seguridad de Java se establece a tres niveles:

- Nivel de seguridad dado por las características del lenguaje, tales como la ausencia de punteros (que evita cualquier error de asignación de memoria) o el "ocultamiento de la información" propio de la programación orientada a objetos, por recordar dos ejemplos ya mencionados.

- Nivel de seguridad dado por el diseño de la VM: La VM de Java posee un **verificador de los byte codes**, que antes de ejecutarlos analiza su formato comprobando que no existen punteros en ellos, que se accede a los recursos del sistema a través de objetos de Java, etc.

Otro elemento constitutivo de la VM es el **cargador de clases**. Una clase es una categoría de objetos utilizados en un programa; cuando se ejecuta un programa en Java, éste llama a determinadas clases a través del cargador de clases. Estas clases pueden provenir de tres lugares distintos, en donde residen en forma de ficheros: del ordenador local, de la red de área local a la que pueda estar conectado el ordenador cliente, o de Internet. En función de la procedencia de las clases, se efectúan

una serie de comprobaciones diferentes y el **gestor de seguridad** de la VM prohíbe los accesos peligrosos.

- Nivel de seguridad dado por la API de Java. El conjunto de métodos y clases que estamos obligados a utilizar cuando programamos en Java para acceder a los recursos del sistema, está definido por la API, y constituye la última barrera defensiva. El diseño de dichos métodos y clases hace que éstos realicen múltiples verificaciones cuando son invocados, de modo que se dificultan los errores (voluntarios o involuntarios).

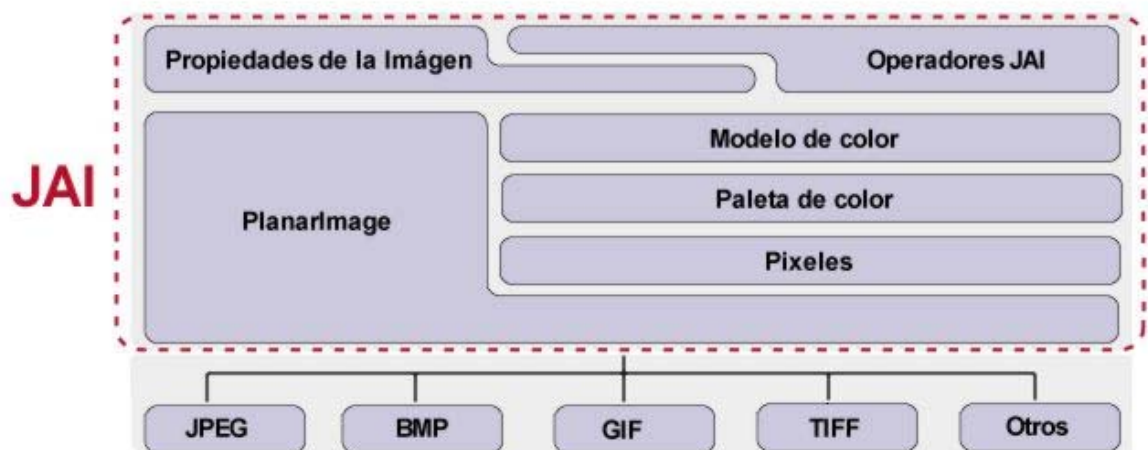
- **Adaptación a redes (en particular, a Internet).** Ya hemos dicho que Java irrumpió en el mercado para potenciar la interactividad en Internet, y también hemos hablado de los **applets**, pequeños programas en Java que se cargan junto con una **página web** desde un servidor, y que son ejecutados (por la VM del navegador del cliente) como una parte de la página web. Además de las ventajas que supone su ejecución local, los applets disponen de una significativa riqueza de recursos y son capaces de realizar **tareas muy complejas** a pesar de su reducido tamaño. Una de las explicaciones de esta sorprendente capacidad es el hecho de que los applets se sirven del propio código del navegador en cuya VM se ejecutan, utilizándolo para tareas tales como presentación gráfica o comunicaciones. Sin embargo, el acceso a las funciones del navegador es totalmente automático y transparente para el programador, que debe limitarse a invocar ciertas funciones de la API de Java; estas invocaciones, interpretadas por el navegador, dan origen a acciones muy complejas. Esta observación es muy importante cuando se discute del **rendimiento** de Java, pues todas estas acciones se realizan en la máquina que está ejecutando el applet, y la rapidez de ejecución de las mismas no depende de que Java sea un lenguaje semiinterpretado (o semicompilado).

Entre las tareas básicas más comunes que suelen realizar los applets, se encuentran: visualizar animaciones en la ventana del navegador; reproducir sonidos; establecer comunicaciones con el servidor del que procede el applet (p. ej. para cargar desde él ficheros de cualquier naturaleza); crear interfaces gráficas con los elementos habituales de los entornos de ventanas (como Macintosh o Windows): menús desplegables, botones, áreas de texto, barras de desplazamiento, etc.; pedir datos al usuario para procesarlos (gestionando eventos como pulsaciones de teclas o acciones con el ratón); etc. No obstante, y como acabamos de decir, es posible programar applets para realizar tareas de enorme complejidad.

9.2 Las bibliotecas JAI de Sun Microsystems

Estas bibliotecas están pensadas para satisfacer las necesidades de cualquier aplicación que manipule imágenes. Son fácilmente extensibles, permitiendo que nuevas operaciones de proceso de imágenes sean añadidas y pasen a estar disponibles para todos los desarrolladores Java que quieren incorporar tratamiento de imágenes en sus aplicaciones y applets.

Las JAI encapsulan los diferentes formatos de imagen y las invocaciones a métodos remotos con un objeto de imagen, permitiendo que un archivo de imagen, un objeto de imagen de red, o una imagen transmitida en tiempo real sean procesadas de la misma forma. De esta manera, gracias a esta biblioteca, contamos con un modelo simple de programación abstrayéndonos de los complejos mecanismos internos del tratamiento de imágenes.



10. DISEÑO DE LA ARQUITECTURA

Observando la arquitectura diseñada en años anteriores, realizamos una división del sistema acorde y una definición de interfaces de las distintas partes del sistema, así como las interfaces con distintos sistemas externos (impresora, portapapeles, configuración de ftp...) con los que nuestra aplicación pudiera tener relación.

Se trató de conseguir un diseño abierto a nuevas funcionalidades, escalable, portable y ampliable, en proyectos posteriores.

A continuación mostramos una relación representativa de las clases principales de la aplicación:

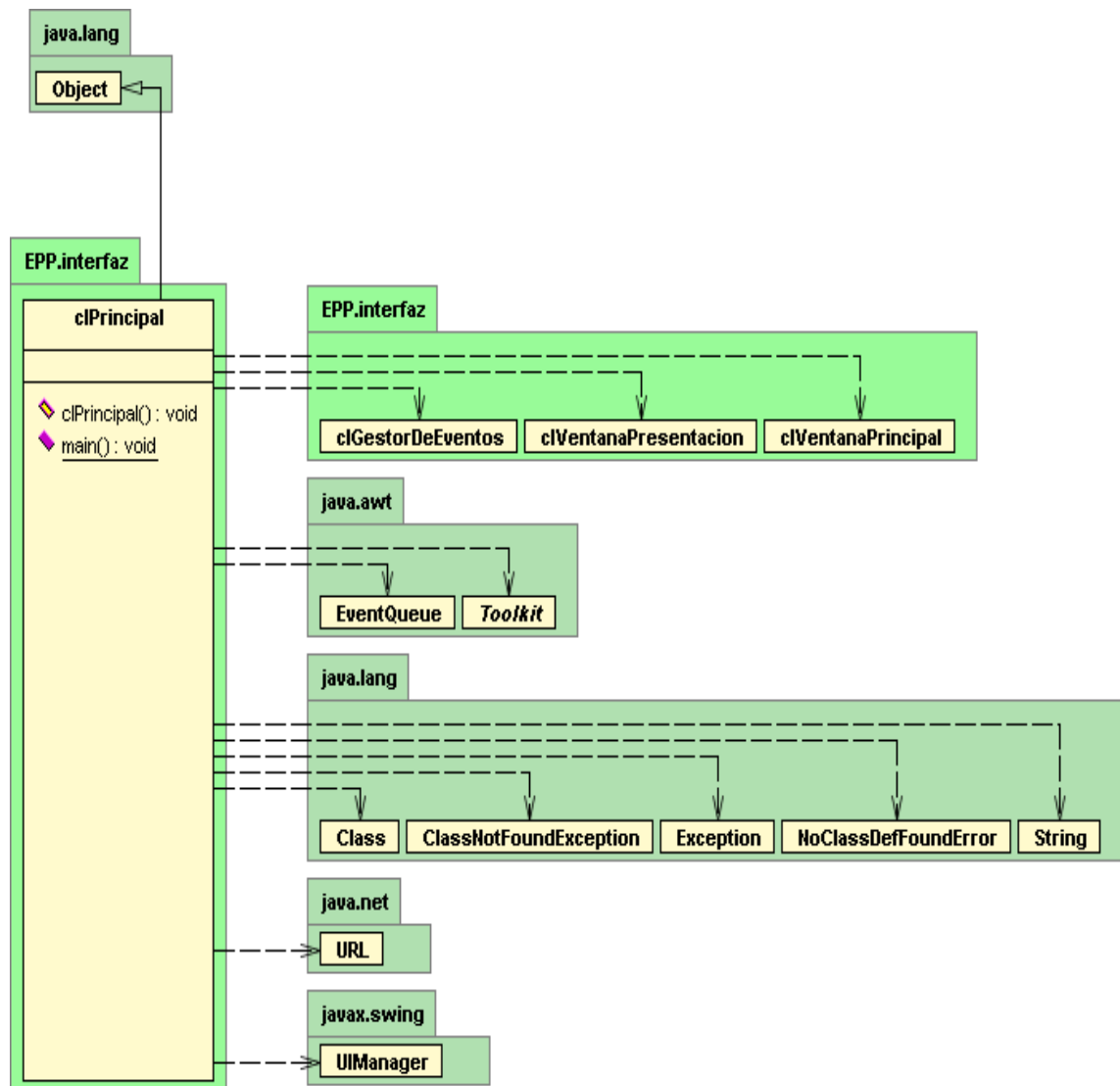


Diagrama UML de cIPrincipal

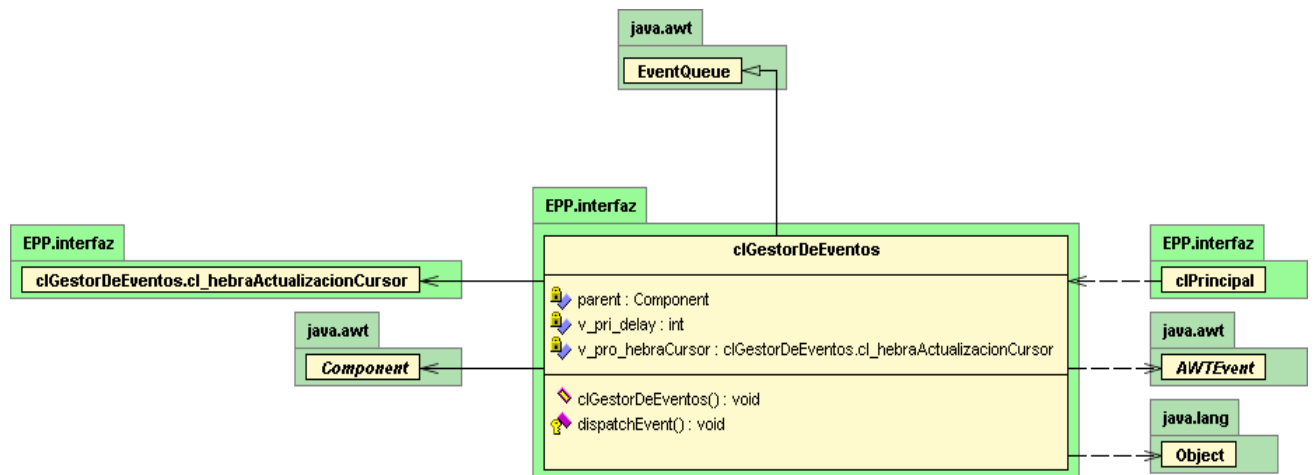


Diagrama UML de `cIGestorDeEventos`

Para consultar el resto de diagramas en UML, de las relaciones entre las distintas clases de la aplicación, puede consultarse en el CD de la memoria en la carpeta de Documentación/docs, en el apartado de UML.

10.1 Modularidad

La modularidad, es la capacidad de poder trabajar de manera independiente y paralela en distintas partes del programa, además de conseguir que el mismo sea ampliable y fácil de comprender. La modularidad es el atributo del software que permite a un programa ser manejable intelectualmente.

Todo ello se ha tratado de mantener a lo largo del desarrollo de nuestro proyecto. Dado que la modularidad consiste en dividir el software en componentes identificables y tratables por separado, nosotros lo conseguimos con los llamados custom events, cuya mayor ventaja es la transparencia. Con el sistema de notificación de eventos ni siquiera necesitamos saber el número de objetos, módulos, etc... "oyentes" (es decir, que tienen una dependencia con el origen del evento).

Cumpliendo los principios básicos de la modularidad, obtenemos con los eventos:

- El uso de objetos plenamente autónomos, que pueden ser usados en diferentes puntos de la aplicación así como en otra aplicación distinta. Es decir, el desarrollo de este tipo de objetos, permite vender componentes por separado o integrar los objetos en un servidor de aplicaciones.

- El uso de objetos con capacidad de relación con su entorno, intercambiando la menor cantidad de datos posible. Con ello se consigue solucionar en parte uno de los grandes problemas del lenguaje Java: la eficiencia.
- El uso de objetos con capacidad de acotar los cambios a eventos puntuales.

10.2 Reusabilidad

La reusabilidad, entendida como la facilidad para usar partes del proyecto en otros proyectos así como en la ampliación del mismo en años posteriores, es la consecuencia directa de la modularidad.

Para ello se han separado los distintos menús y formularios en distintas clases, se han creado interfaces e implementado clases abstractas para facilitar la ampliación de la aplicación.

Esta técnica se ha seguido, por ejemplo, en el desarrollo de los formularios de la aplicación. Implementamos una primera clase abstracta (clFormularioBase) de la que derivan todos los formularios de entrada de datos a la aplicación. Igualmente seguimos este esquema de desarrollo a la hora de implementar las clases referentes a filtros, bordes y diferentes plug-ins.

Los conceptos descritos pueden verse reflejados en los diagramas del Anexo E, "Análisis del sistema".

11. DISEÑO DETALLADO

En este apartado tratamos de diseñar e implementar los distintos componentes obtenidos en la fase general del diseño descrita en el punto "10. DISEÑO DE LA ARQUITECTURA", así como de realizar las pruebas unitarias de cada componente y su posterior integración.

Hemos prescindido de las facilidades para el diseño de formularios que ofrece JBuilder. De esta manera hemos conseguido una interfaz más flexible y dinámica reduciendo el espacio necesario para la ejecución de la aplicación.

En la fase de implementación, contamos con el apoyo de las librerías JAI, para realizar diferentes operaciones sobre los distintos estándares de almacenamiento de imágenes, si bien muchas funciones ofrecidas por la aplicación hemos tenido que desarrollarlas por completo.

11.1 JAI en nuestra aplicación

Dentro de nuestra aplicación, hemos empleado diversas bibliotecas de JAI, las cuales fueron explicadas en el punto “9.2 Las bibliotecas JAI de Sun Microsystems”, de esta misma memoria.

Las bibliotecas más usadas han sido principalmente las referidas a las clases de `PlanarImage` y `TiledImage`, así como las operaciones de bajo nivel que proporciona, divididas en nueve categorías principales, que se detallan a continuación:

- Operadores de punto (Point Operators): Las operaciones de punto transforman una imagen de entrada en una imagen de salida de manera que cada píxel de la imagen de salida depende solamente del correspondiente píxel de la imagen de entrada.
- Operadores de área (Area Operators): Los operadores de área realizan transformaciones geométricas, que resultan en el reposicionado de píxeles dentro de una imagen. Utilizando transformaciones matemáticas, se obtienen las coordenadas de los píxeles en la imagen de salida a partir de las coordenadas x e y de la imagen de entrada. Hay dos tipos básicos de operaciones de área: Lineares y no lineares. Las operaciones lineares incluyen la traslación, la rotación, y el escalado. Las operaciones no lineares, también conocidas como *warping transformations*, introducen curvaturas y ángulos a la imagen procesada.
- Operadores geométricos (Geometric Operators): Los operadores geométricos permiten modificar la orientación, tamaño y forma de una imagen.
- Operadores de cuantificación de color (Color Quantization Operators): La cuantificación de color, también conocida como dithering, se emplea para efectuar transformaciones de color sobre los píxeles de la imagen de entrada.
- Operadores de fichero (File Operators): Los operadores de fichero se emplean para leer o escribir ficheros de imagen.
- Operadores de frecuencia (Frequency Operators): Los operadores de frecuencia se usan para descomponer una imagen desde su forma de dominio espacial a una forma del dominio de la frecuencia. También se proporcionan operadores para realizar la transformación inversa, en la cual la imagen se convierte desde el dominio de la frecuencia de nuevo al dominio espacial.

- Operadores estadísticos (Statistical Operators): Los operadores estadísticos proporcionan los medios para analizar el contenido de una imagen.
- Operadores de extracción de bordes (Edge Extraction Operators): Los operadores de extracción de bordes permiten el realzado de los bordes de una imagen. El realzado de bordes reduce una imagen para mostrar solamente sus detalles de bordes. El realzado de bordes se implementa a través de filtros espaciales que detectan una específica pendiente de brillo de píxel en un grupo de píxeles de una imagen. Una pendiente de brillo acusada indica la presencia de un borde.
- Otros operadores (Miscellaneous Operators): Aquí se engloban otros operadores que no encajan en ninguna de las categorías previas.

Empleando estos operadores, la filosofía que pretende seguir la API de JAI es proporcionar un sistema que permita evitar tratar directamente el formato interno de la imagen, es decir, el formato de los píxeles, el modelo de color y la paleta de color.

Esta idea, que puede parecer la aproximación más adecuada a la hora de enfrentarnos al diseño de una aplicación de manipulación de imágenes, si bien resulta apropiada a la hora de manipular o realizar operaciones básicas, como las anteriormente comentadas, demuestra sus carencias, complejidad e ineficiencia al intentar implementar operaciones más complejas, ya que tanta encapsulación y aislamiento, perjudican a una aplicación de este tipo que busca tratar con el formato interno en muchas ocasiones. Este es el motivo que ha dado lugar a que en muchos casos, nos hayamos visto forzados a renunciar a uso de operadores de JAI y programar nuestras propias funciones.

Otro problema que hemos encontrado a la hora de emplear la API de JAI es el hecho de que algunos de los operadores ofrecidos funcionan incorrectamente, o directamente no funcionan, dependiendo del formato de la imagen y otras propiedades de la imagen (como por ejemplo, regiones transparentes), lo cual se ve agravado por el hecho de que la API de JAI se entregue “as is”, es decir sin ningún tipo de garantía ni de mantenimiento, por lo que en algunos casos hemos debido recurrir de nuevo a la definición de nuevas clases con operadores en teoría proporcionados por las JAI.

11.2 Sencillez de manejo

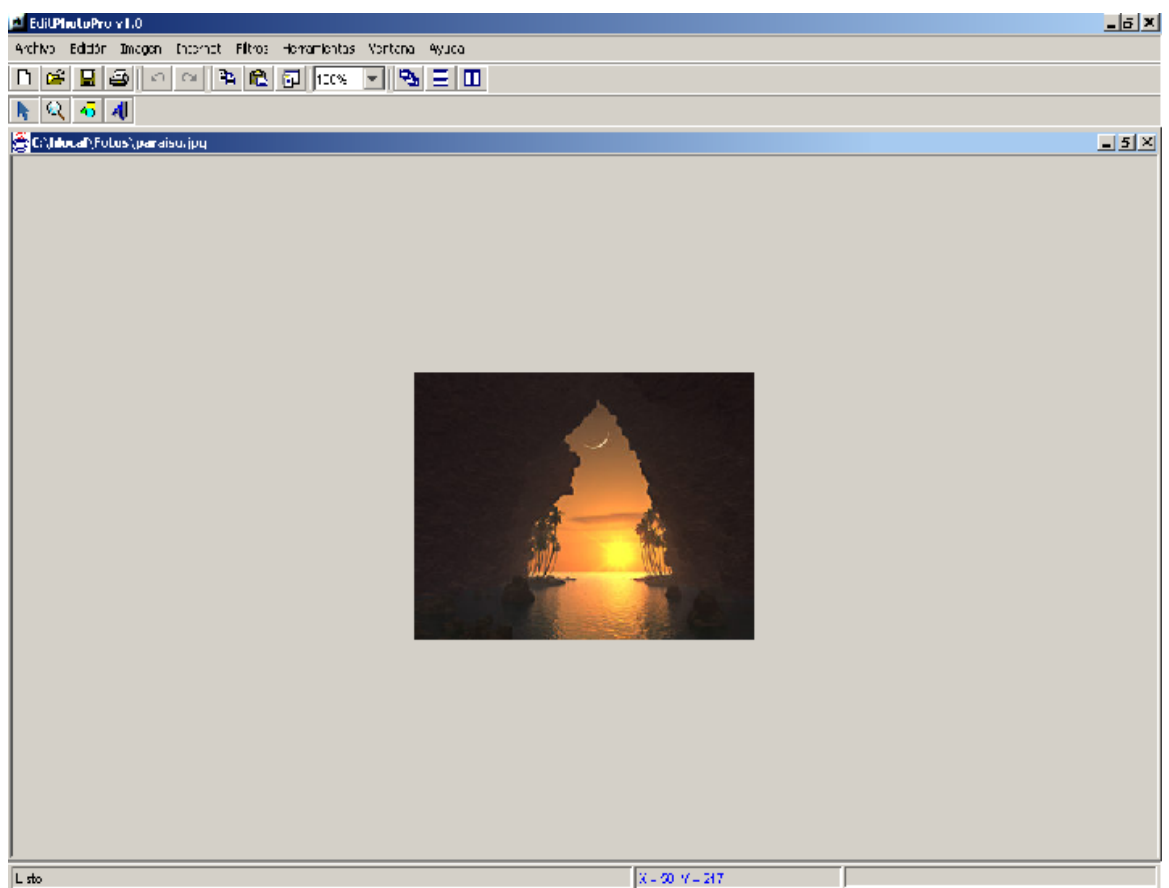
Como hemos tratado de dejar patente, nuestra aplicación está orientada a fotógrafos principiantes y a profesionales, a través de un entorno

amigable y fácil de manejar, por usuarios, que no necesariamente, tengan conocimientos de informática.

La interacción con la aplicación se lleva a cabo de una manera intuitiva, lógica y similar a otras herramientas existentes en el mercado, las cuales están descritas en el punto “2.3 Antecedentes” de esta misma memoria. Como podemos observar más abajo, el acceso a las diferentes funcionalidades ofrecidas por Edit Photo Pro v1.0, se puede realizar a través de una barra de Menús o de una barra de Herramientas más intuitiva. Así mismo, se incluyen facilidades de acceso rápido mediante teclado.

El interfaz conseguido es fácilmente comprensible por cualquier tipo de usuario, tanto si está familiarizado con el uso de programas de retoque fotográfico como si no.

El entorno gráfico creado, nos recuerda a las aplicaciones en las que nos hemos basado, como puede observarse a continuación:



Además, la aplicación se acompaña de un manual de instalación y un manual de usuario, que pueden ser consultados en el Anexo H, “Instalación” y en el Anexo I, “Manual de usuario”, respectivamente. Estos manuales se han desarrollado, de manera que sirvan de ayuda a cualquier usuario con conocimientos informáticos escasos, para que pueda instalar y manejar la aplicación de manera sencilla y amigable.

12. PRUEBAS DE SISTEMA

En un principio las pruebas las llevamos a cabo los integrantes del proyecto. Tras este primer “testeo” corregíamos los diferentes fallos encontrados, con el fin de poder realizar una segunda fase de pruebas, desarrollada por personas ajenas a nuestro proyecto.

Gracias a esta segunda fase, pudimos mejorar distintos aspectos de la aplicación, tales como generar plantillas para los álbumes que más tarde subiríamos a Internet.

13. IMPLANTACIÓN Y CIERRE DEL PROYECTO

Dado el ámbito del proyecto, la fase de implantación queda fuera de nuestro alcance.

En cambio, si podemos hablar de un primer cierre del proyecto. El producto final es completo y cerrado, lo que no implica que no pueda servir como base a futuras ampliaciones, ya que este tipo de aplicaciones ofrece grandes posibilidades en cuanto a un posible trabajo futuro.

Para obtener información más detallada sobre las posibles extensiones del producto consultar el punto “14. TRABAJO FUTURO” de esta misma memoria.

14. TRABAJO FUTURO

14.1 Procesamiento de imágenes avanzado

En este apartado, se describen algunos tratamientos propuestos como posible ampliación para proyectos futuros. Las siguientes funcionalidades descritas podrían ser requeridas para una especialización del producto de cara a su uso por profesionales del sector de la fotografía y el diseño gráfico.

Registro de imágenes avanzado

El *registro de imágenes* es un proceso que consiste en alinear dos o más imágenes de la misma escena. Normalmente, se toma como referencia una imagen, llamada imagen de base, con la que se comparan las otras imágenes, llamadas imágenes de entrada. El objetivo del registro de imágenes es alinear las imágenes de entrada con la imagen de base mediante la aplicación de una transformación espacial sobre aquéllas.

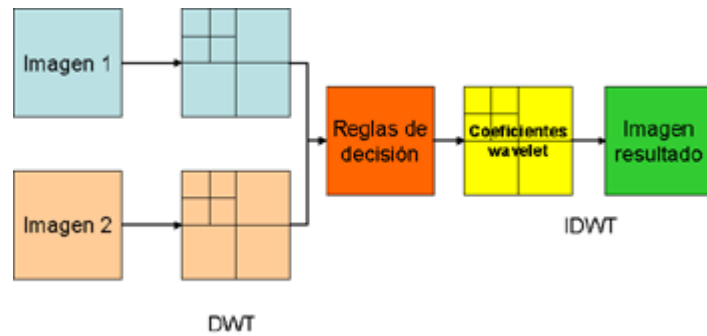


La imagen de color pequeña se ha sobrepuesto a la imagen grande después de determinar en qué lugar se debe colocar.

Fusión de imágenes

La *fusión de imágenes*, es una operación que se realiza de múltiples formas, pero que se describirá brevemente en la modalidad de nivel de píxel. Para ello, un pre-requisito fundamental consiste en la alineación de las imágenes de entrada píxel a píxel, lo que conlleva otras implicaciones como por ejemplo la identidad del tamaño matricial.

Uno de los varios mecanismos de fusión se basa en la transformada wavelet discreta. El siguiente diagrama describe un algoritmo muy simple y al mismo tiempo muy general para llevar a la práctica la fusión, en el supuesto de que se cumplan los requisitos básicos antes mencionados.



La selección de las reglas de decisión sobre la combinación de los coeficientes wavelet varía en función del resultado que se desea obtener. Puesto que los coeficientes de mayor módulo corresponden a los bordes y cambios más relevantes en las imágenes, una posibilidad consiste en seleccionar, para cada píxel, el de módulo máximo entre los coeficientes correspondientes a las distintas imágenes de entrada.

Se ilustra el algoritmo mediante dos imágenes de la catedral de Santiago de Compostela parcialmente degradadas en segmentos distintos. La tercera corresponde al resultado de la fusión.





14.2 Ampliación de funcionalidades

Debido a que el tratamiento de imágenes digitales es un campo muy extenso, y la consabida limitación temporal del proyecto, hemos limitado las funcionalidades incluidas en el producto final. No por ello, se han incumplido los requisitos mínimos que nos marcamos al principio.

No nos centramos en desarrollar gran número de operaciones sobre imágenes, debido a que nuestro objetivo principal era el de conseguir organizar nuestras fotos en álbumes en un formato compatible con cualquier plataforma (los álbumes están generados en html), así como optar por compartir nuestros álbumes en la red.

Sin embargo, las posibilidades de ampliación son casi infinitas, ya que muchas funcionalidades han quedado fuera del proyecto, y a que el tratamiento de imágenes es un campo en continua expansión.

No hay más que recurrir a cualquier aplicación comercial ya mencionada en el punto “2.3 Antecedentes”, para ver el gran número de mejoras que pueden realizarse sobre nuestra aplicación.

- Ampliaciones de tratamiento de imágenes: ya que la aplicación se orientó en un principio hacia los profesionales de la fotografía, una posible mejora sería incluir efectos “artísticos”, como por ejemplo, dar aspecto de acuarela, cambiar la textura de la imagen, etc.
- Ampliaciones de creación de álbumes: una primera ampliación podría ser el ampliar el número de plantillas sobre las que crear los álbumes, dando a escoger al usuario igualmente la posibilidad de personalizar una propia.
- Ampliaciones de actualización en web: una mejora considerable que aporta gran versatilidad a la aplicación, sería la posibilidad de configurar el ftp al que se subirán los álbumes creados. De esta manera, el usuario solamente debería aportar los datos de dirección del ftp, nombre de usuario y contraseña una vez y utilizar esta nueva configuración siempre que quisiera.

Por supuesto, estas son solamente algunas de las sugerencias que nos parecen más importantes a la hora de ampliar el proyecto.

15. BIBLIOGRAFÍA

- LIBROS

- Pajares, Gonzalo; De la Cruz, Jesús M. "Visión por Computador. Imágenes Digitales y Aplicaciones" Rama, 2001.
- Fu, K.S., González, R.C., Lee, C.S.G. "Robótica.Control, Detección,Visión e Inteligencia". McGraw-Hill, 1988.
- Maravall, D. "Reconocimiento de formas y visión artificial". Rama, 1993.
- Douglas A. Lyon, Douglas A. Lyons. "Image Processing in Java". A Lyon Book.
- Nick Efford. "Digital Image Processing". School of Computer Studies, University of Leeds.
- Romaniello, Steve, "*Photoshop 7*", Anaya Multimedia, Madrid, 2003.
- J. D. Foley, A. Van Dam, S. K. Freiner, J .F. Hughes. "Computer Graphics Principles and Practice". Addison-Wesley, Reading: MA, 1993.

- DIRECCIONES WEB

- Sun Microsystems: www.sun.com
- Java™ Advanced Imaging API: java.sun.com/products/java-media/jai/index.html
- Borland: www.borland.com
- Java en castellano: www.programacion.com/java/
- Librerías ftpBean: www.geocities.com/SiliconValley/Code/9129/javabean/ftpbean
- Otras webs de interés: www.adictosaltrabajo.com/tutoriales.php?pagina=nsis
www.javahispano.org
www.sourceforge.net
www.google.es

16. ANEXOS

16.1 ANEXO A: Estándar de Documentación

16.2 ANEXO B: Gestión de Configuración

16.3 ANEXO C: Análisis de riesgos

16.4 ANEXO D: Especificación de Requisitos

16.5 ANEXO E: Análisis del sistema

16.6 ANEXO F: Cómo añadir elementos a la interfaz

16.7 ANEXO G: Cómo añadir una nueva transformación

16.8 ANEXO H: Instalación

16.9 ANEXO I: Manual de usuario

16.1 ANEXO A: **Estándar de** **documentación**

El formato de letra elegido es Arial. El esquema básico seguido es el siguiente:

1. TÍTULO PRIMERO

1.1 Subtítulo primero

Descripción del apartado.

- APARTADO PRIMERO

Descripción del apartado.

- APARTADO SEGUNDO
- ...

1.2 Subtítulo segundo

Descripción del apartado

- Apartado primero
- Apartado segundo

...

2. TÍTULO SEGUNDO

3. TÍTULO TERCERO

4. ...

Al incluir fotos o cuadros de texto en un documento, éstos irán centrados en el documento con el pie de foto en letra Arial de tamaño 10 en cursiva, igualmente centrado.

16.2 ANEXO B: **Gestión de** **configuración**

16.3.1 INTRODUCCIÓN

Este documento describe las distintas entidades que formarán parte de la gestión de configuración del proyecto y los procesos que les serán aplicados. También detalla las distintas herramientas que serán usadas para llevar a cabo este proceso.

16.3.2 PROCESO DE ENVÍO DE ENTIDADES A GC

El envío de entidades para gestión de configuración se realizará a través del siguiente grupo de distribución:

sistemas_inf2004@yahoogroups.com

El formato del asunto será el siguiente:

<nombre del documento>

16.3.3 ENTIDADES SOMETIDAS A GESTIÓN DE CONFIGURACIÓN

A continuación se detallan las distintas entidades del proyecto que serán sometidas a gestión de configuración.

16.3.3.1 Código fuente de la aplicación

El código fuente de la aplicación está escrito en Java y el formato de su contenido está descrito en el documento de estándar de código.

16.3.3.2 Plantillas y documentos

Los documentos generados para el proyecto seguirán el siguiente formato:

Fecha-<nombre del documento>.

Además, todos estos documento deberán seguir el estándar de documentación del proyecto.

16.3.4 CONTROL DE VERSIONES

Las versiones del código tendrán el siguiente formato: v1.0.

El primer número identifica la fase en que se encuentra el proyecto. El segundo identificará cambios relevantes dentro de cada fase.

16.3.5 HERRAMIENTAS

Las herramientas que serán usadas en el proceso de gestión de configuración se detallan a continuación:

Listas de correo

Este sistema permite la comunicación por correo electrónico entre los miembros del grupo.

A través de la siguiente dirección:

sistemas_inf2004@yahoogroups.com

cada miembro del proyecto puede obtener las últimas versiones de código y de documentación.

Java

El proyecto se está desarrollando con JBuilder 9 incluyendo las bibliotecas Jai de Sun Microsystems.

16.3 ANEXO C: **Análisis de riesgos**

16.4.1 IDENTIFICACIÓN DE RIESGOS

Tipos	Posibles riesgos	Afectan a
Riesgos tecnológicos	<ul style="list-style-type: none"> - Limitaciones que surjan del lenguaje Java - Fallo en el servidor de listas de distribución - Descapacitación de hardware, es decir, que no contemos con los permisos necesarios, en el laboratorio, espacio en disco, memoria... - Componentes software que reuses podrían tener defectos de implementación que limiten su funcionalidad - Extravío de documentos o de archivos 	<ul style="list-style-type: none"> - Proyecto y producto - Proyecto - Proyecto - Producto - Proyecto y producto
Riesgos de personal	<ul style="list-style-type: none"> - Enfermedad o abandono de un miembro del grupo - Disputas entre componentes del grupo - Los miembros carezcan de formación adecuada en algunos campos, o del conocimiento de uso de ciertas herramientas - Componente del grupo se descuelga y le cuesta coger de nuevo el ritmo 	<ul style="list-style-type: none"> - Proyecto - Proyecto - Proyecto y producto - Proyecto
Riesgos de la organización	<ul style="list-style-type: none"> - Cambio de profesor del proyecto - Cambio en la forma de evaluación - Descoordinación que conlleve a que un mismo trabajo sea realizado por varios componentes del grupo 	<ul style="list-style-type: none"> - Proyecto - Proyecto - Proyecto
Riesgos de requisitos	<ul style="list-style-type: none"> - Cambio en los requisitos iniciales - Retrasos en la especificación - Adelanto en la fecha de entrega del producto - Incumplimiento de requisitos fijados por causa de falta de tiempo o desconocimiento de su forma de realización - Diferencia entre la interpretación de los requisitos pactados inicialmente por el equipo y los resultados finales 	<ul style="list-style-type: none"> - Proyecto y producto - Proyecto y producto - Proyecto - Proyecto y producto - Producto
Riesgos de estimación	<ul style="list-style-type: none"> - Nos hemos quedado cortos a la hora de estimar el tiempo para desarrollar el software - Hemos estimado a la baja el tamaño de la aplicación en disco - La aplicación en ejecución crece más en memoria que lo que se estimó en un principio - La aplicación consume más tiempo en ejecutar ciertas funcionalidades del que se fijó - Fallo en la estimación inicial del tiempo que cada uno de los componentes podíamos dedicar al desarrollo 	<ul style="list-style-type: none"> - Proyecto y producto - Proyecto y producto - Proyecto y producto - Proyecto y producto - Proyecto y producto

Riesgos de herramientas	<ul style="list-style-type: none"> - Problemas con el manejo de herramientas nuevas, de las que carecemos de un conocimiento para su uso eficiente - Incompatibilidades de horarios para hacer uso de los laboratorios de la facultad, donde se encuentran las herramientas que necesitaremos para el desarrollo 	<ul style="list-style-type: none"> - Proyecto - Proyecto
-------------------------	--	--

16.4.2 ANÁLISIS DE RIESGOS

Posibles riesgos	Probabilidad	Efecto
- Limitaciones que surjan del lenguaje Java	- Moderada	- Tolerable
- Fallo en el servidor de listas de distribución	- Baja	- Catastrófico
- Descapacitación de hardware, es decir, que no contemos con los permisos necesarios, en el laboratorio, espacio en disco, memoria...	- Alta	- Catastrófico
- Componentes software que reusemos podrían tener defectos de implementación que limiten su funcionalidad	- Alta	- Tolerable
- Extravío de documentos o de archivos	- Baja	- Catastrófico
- Enfermedad o abandono de un miembro del grupo	- Moderada	- Serio
- Disputas entre componentes del grupo	- Moderada	- Serio
- Los miembros carezcan de formación adecuada en algunos campos, o del conocimiento de uso de ciertas herramientas	- Alta	- Tolerable
- Componente del grupo se descuelga y le cuesta coger de nuevo el ritmo	- Moderada	- Tolerable
- Cambio de profesor del proyecto	- Baja	- Serio
- Cambio en la forma de evaluación	- Baja	- Serio
- Descoordinación que conlleve a que un mismo trabajo sea realizado por varios componentes del grupo	- Moderada	- Serio
- Cambio en los requisitos iniciales	- Alta	- Tolerable
- Retrasos en la especificación	- Moderada	- Serio
- Adelanto en la fecha de entrega del producto	- Baja	- Catastrófico
- Incumplimiento de requisitos fijados por causa de falta de tiempo o desconocimiento de su forma de realización	- Alta	- Serio
- Diferencia entre la interpretación de los requisitos pactados inicialmente por el equipo y los resultados finales	- Baja	- Tolerable
- Nos hemos quedado cortos a la hora de estimar el tiempo para desarrollar el software	- Alta	- Serio
- Hemos estimado a la baja el tamaño de la aplicación	- Alta	- Serio

en disco

- La aplicación en ejecución crece más en memoria que lo que se estimó en un principio - Alta - Serio
- La aplicación consume más tiempo en ejecutar ciertas funcionalidades del que se fijó - Alta - Serio
- Fallo en la estimación inicial del tiempo que cada uno de los componentes podíamos dedicar al desarrollo - Moderada - Catastrófico
- Problemas con el manejo de herramientas nuevas, de las que carecemos de un conocimiento para su uso eficiente - Muy alta - Tolerable
- Incompatibilidades de horarios para hacer uso de los laboratorios de la facultad, donde se encuentran las herramientas que necesitaremos para el desarrollo - Baja - Catastrófico

16.4.3 PLANIFICACIÓN DE RIESGOS

Posibles riesgos	Estrategia a seguir
Limitaciones que surjan del lenguaje Java	Aproximar lo máximo posible si es prioritario y descartarlo si no lo es
Fallo en el servidor de listas de distribución	Uso de mails personales o en caso extremo quedar personalmente e intercambiar la información
Descapacitación de hardware, es decir, que no contemos con los permisos necesarios, en el laboratorio, espacio en disco, memoria...	Cambiar de puesto, hablar con el responsable o descartar las funcionalidades que hagan uso de estos requisitos
Componentes software que reusemos podrían tener defectos de implementación que limiten su funcionalidad	Buscar nuevos componentes que sean compatibles y adaptar lo existente a ellos
Extravío de documentos o de archivos	Rehacer el documento
Enfermedad o abandono de un miembro del grupo	Repartir el trabajo entre el resto
Disputas entre componentes del grupo	Resolverlas hablándolo entre los componentes o avisar al profesor en caso extremo
Los miembros carezcan de formación adecuada en algunos campos, o del conocimiento de uso de ciertas herramientas	Si alguno de los miembros tiene la formación adecuada, hará una pequeña conferencia para explicárselo a los demás,

	si no, un miembro se formará y hará lo mismo
Componente del grupo se descuelga y le cuesta coger de nuevo el ritmo	Reunión con todos los miembros para poner al día al componente
Cambio de profesor del proyecto	Rehacer desde cero el proyecto en caso de que el profesor lo requiera
Cambio en la forma de evaluación	Adaptarse a la nueva manera
Descoordinación que conlleve a que un mismo trabajo sea realizado por varios componentes del grupo	Reunión para escoger la mejor versión y redistribuir el trabajo para evitar que vuelva a suceder
Cambio en los requisitos iniciales	Adaptar la versión actual para que cumpla los nuevos requisitos con el mínimo cambio posible
Retrasos en la especificación	Aplicar más medios para cumplir el plazo
Adelanto en la fecha de entrega del producto	Aplicar más medios para cumplir el plazo
Incumplimiento de requisitos fijados por causa de falta de tiempo o desconocimiento de su forma de realización	Adaptar la versión actual para que cumpla los nuevos requisitos con el mínimo cambio posible
Diferencia entre la interpretación de los requisitos pactados inicialmente por el equipo y los resultados finales	Convocar reunión con moderador para aclarar la situación
Nos hemos quedado cortos a la hora de estimar el tiempo para desarrollar el software	Redefinir el alcance. Elegir por prioridades sacrificando los requisitos de más baja prioridad
Hemos estimado a la baja el tamaño de la aplicación en disco	Optimizar lo máximo posible el tamaño de la aplicación y en caso de imposibilidad, aumentar los requisitos de hardware
La aplicación en ejecución crece más en memoria que lo que se estimó en un principio	Optimizar lo máximo posible el tamaño de la aplicación y en caso de imposibilidad, aumentar los requisitos de hardware
La aplicación consume más tiempo en ejecutar ciertas funcionalidades del que se fijó	Optimizar lo máximo posible el tamaño de la aplicación y en caso de imposibilidad, aumentar los requisitos de hardware

Fallo en la estimación inicial del tiempo que cada uno de los componentes podíamos dedicar al desarrollo Redefinir el alcance

16.4.4 SEGUIMIENTO DE RIESGOS

Tipos	Posibles indicadores
Riesgos tecnológicos	Fallos del prototipo en las pruebas iniciales, tiempos demasiado malos o fallos en el servidor
Riesgos de personal	Malestar entre los miembros del equipo, baja moral, dejadez por parte de algunos de los miembros del equipo
Riesgos de la organización	Mala sincronización entre los miembros del equipo a la hora de entregar el trabajo, mala gestión de plazos de entrega, falta de organización por parte de los analistas encargados del proyecto
Riesgos de requisitos	Mala especificación de requisitos, al profesor no le convence el trabajo realizado
Riesgos de estimación	Incumplimiento de los plazos de entrega establecidos en principio
Riesgos de herramientas	Las herramientas elegidas inicialmente para el desarrollo no cubren nuestras necesidades o no hemos podido hacer uso de ellas por incompatibilidad con los horarios de los laboratorios de la facultad

16.4 ANEXO D: Especificación de requisitos

Los **requisitos funcionales** de la aplicación son:

- Lectura de imágenes desde archivo en diferentes formatos: GIF, JPEG, TIFF, PNG, PNM y BMP.

- Escritura de imágenes a archivo en diferentes formatos: JPEG, TIFF, PNG, PNM y BMP.
- Control de cambios, esto es, debe permitir deshacer y rehacer.
- Uso de portapapeles local: Cortar, copiar y pegar.
- Transformaciones entre formatos gráficos: RGB, HSI y escala de grises.
- Transformaciones afines sobre imágenes: rotación, inclinado, traslación y escalado.
- Herramientas de análisis de la imagen como:
 - Obtención de los canales de la imagen.
 - Cálculo y visualización del histograma.

Se pretende que la aplicación tenga gestión de capas, entendiendo por capa una sección flotante de la imagen y siendo la imagen la superposición de todas sus capas.

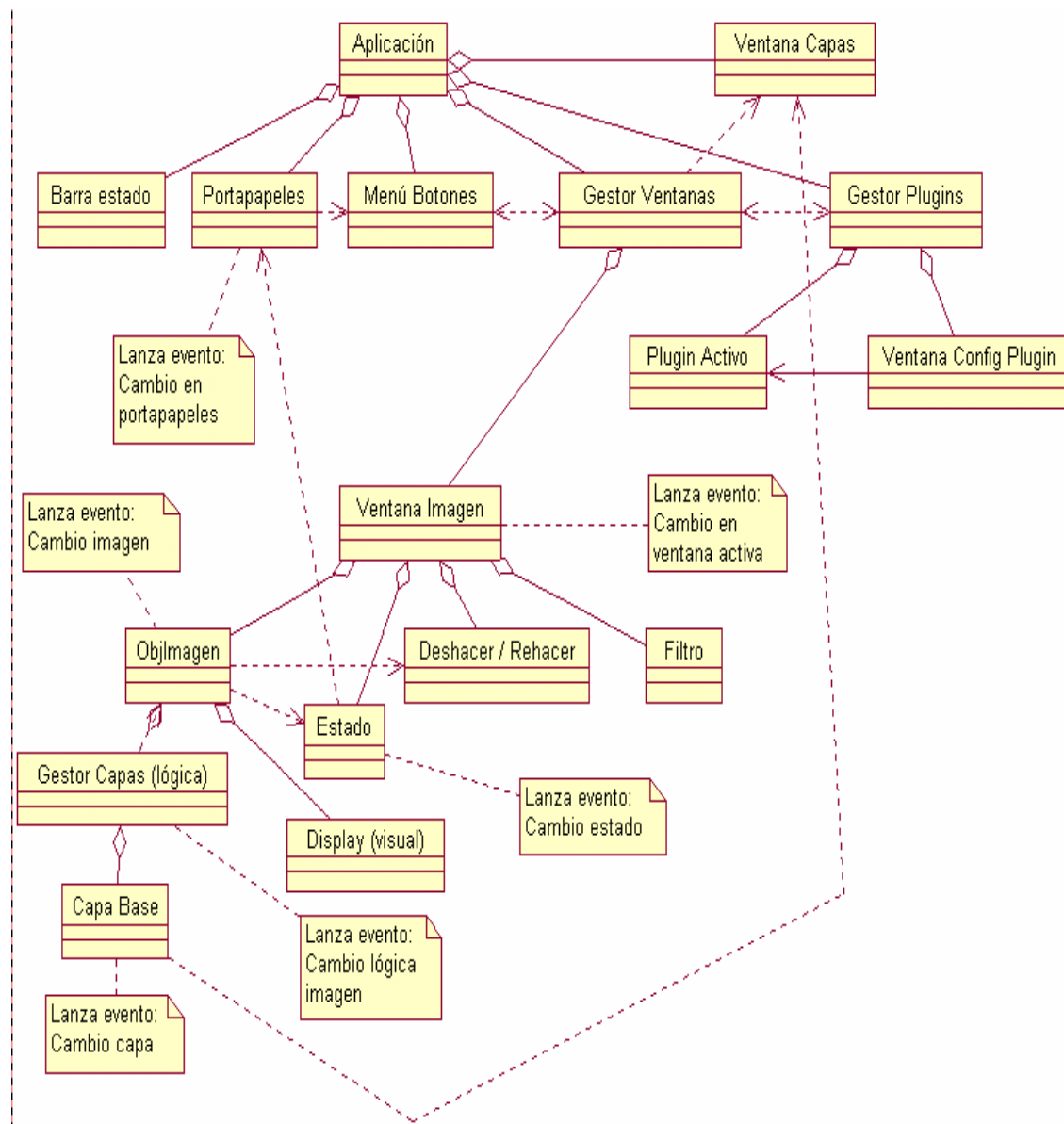
Los **requisitos no funcionales** de la aplicación son:

- Modularidad: El sistema debe ser altamente modular para garantizar la adición de nuevas funcionalidades para manipular imágenes con el menor coste posible.
- Trabajo fluido con un directorio de imágenes.
- Multiplataforma. Este requisito queda satisfecho al haber elegido Java como lenguaje de programación para la aplicación, ya que su uso garantiza el correcto funcionamiento de nuestra herramienta en la totalidad de plataformas de funcionamiento actuales.
- Herramienta de fácil uso y comprensión por parte del usuario. Se pretende implementar un interfaz gráfico similar a los programas de tratamiento de gráficos disponibles en el mercado (Paint Shop Pro, Corel Draw, Adobe Photoshop). Así, cualquier usuario medianamente familiarizado con estas herramientas puede manejar la aplicación sin necesidad de leer ningún manual. En el caso de que el usuario de la aplicación no haya manejado nunca herramientas de este tipo, nuestra herramienta contará con las ayudas necesarias para que no encuentre ningún problema a la hora de familiarizarse con ella.

La aplicación será además sensible al contexto, de forma que se habilitan las opciones de menú que es posible realizar en cada momento, ayudando así al usuario en la comprensión de los pasos a realizar al trabajar con imágenes.

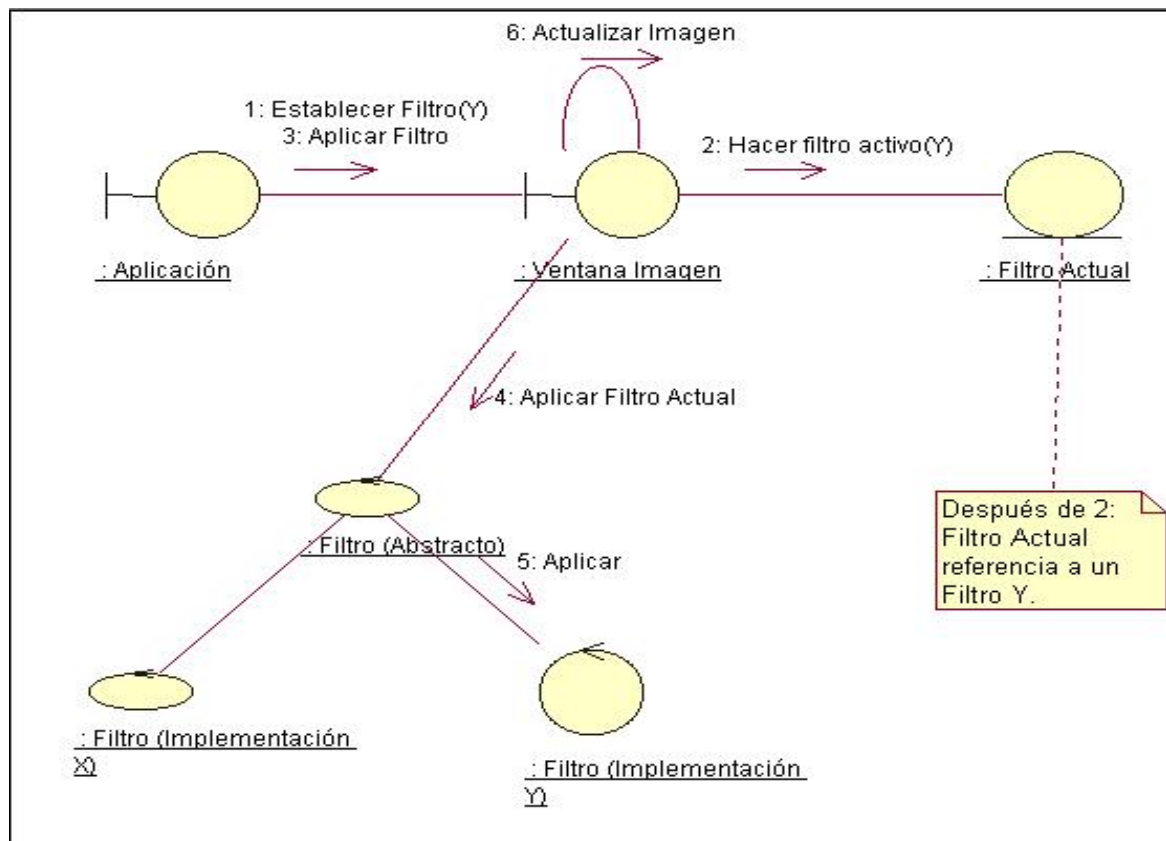
16.5 ANEXO E: **Análisis del Sistema**

16.5.1 ARQUITECTURA DEL SISTEMA

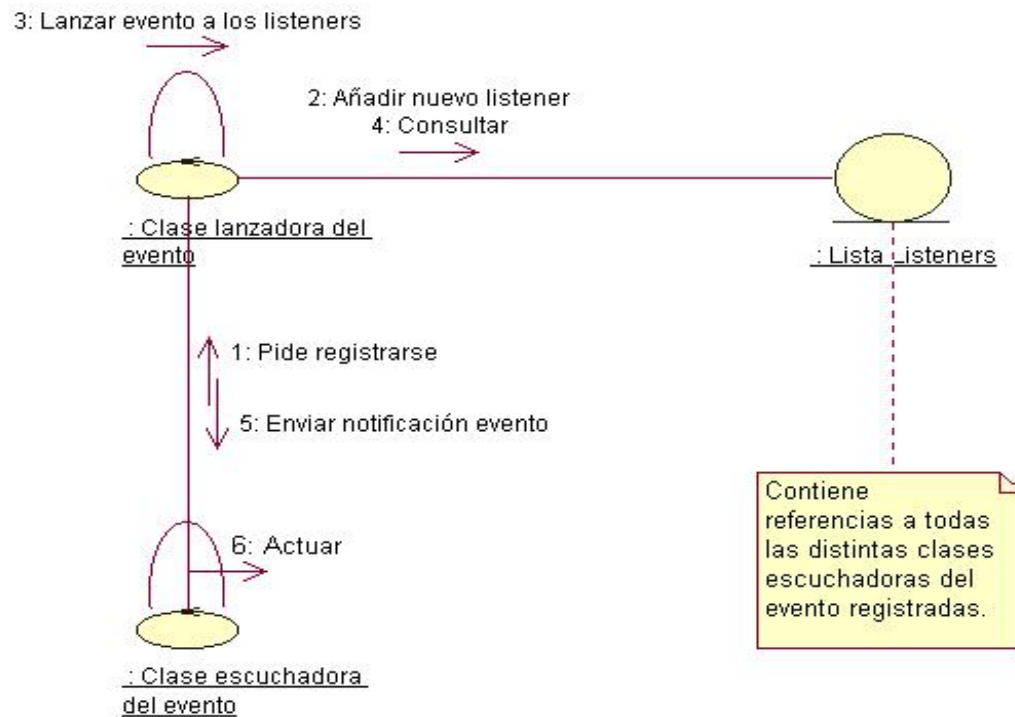


Todos los menús de la aplicación se relacionan de la misma manera que el menú botones.

16.5.2 APLICACIÓN DE UN FILTRO CUALQUIERA



16.5.3 CUSTOM EVENTS (EVENTOS PERSONALIZADOS)



16.6 ANEXO F:

Cómo añadir elementos a la interfaz

Con el objeto de minimizar la cantidad de errores que pudiera provocar el usuario al interactuar con la aplicación, la interfaz permite hacer sólo aquello que tiene sentido en cada momento. De este modo, si no se está procesando ninguna imagen en la aplicación por ejemplo, no se permite guardar, imprimir, aplicar filtros, etc.

Para ello hemos hecho que todos los elementos de la interfaz (menús, botones, ítems y herramientas) sean sensibles tanto a los cambios de ventana dentro de la aplicación como a los cambios en las imágenes contenidas en las ventanas. Esto lo hemos conseguido con la creación del evento personalizado `clEventoVentanaActivada` y la interfaz, en el sentido de *esqueleto de clase Java*, `cl_abs_listenerVentanaActiva`.

Los eventos los lanza el gestor de ventanas cuando se cambia de ventana o se realiza una modificación en una de ellas, y contiene la información relevante que precisan los elementos de la interfaz para saber si deben habilitarse o deshabilitarse.

De este modo, haciendo que los elementos de la interfaz implementen un solo método, el definido en `cl_abs_listenerVentanaActiva`, y declararlos como oyentes del gestor de ventanas para recoger los eventos, conseguimos que modifiquen dinámicamente su estado en función de la información contenida en los mismos.

Insertar un nuevo menú en la barra de menús.

Visto lo anterior, para añadir un nuevo menú a la barra de menús lo único que hay que hacer es:

- 1.- Crear una subclase de `JMenu`, la clase Java que implementa los menús, que implemente la interfaz `cl_abs_listenerVentanaActiva`.
- 2.- Declarar en `clVentanaPrincipal`, la ventana que contiene la aplicación, el nuevo menú como oyente del gestor de ventanas.

Insertar un nuevo ítem en un menú.

Se hace igual que en cualquier otra aplicación Java con la salvedad de que se debe introducir en el método de configuración del menú (el que habilita y deshabilita sus elementos) el código para configurar el nuevo ítem.

Insertar un nuevo botón en la barra de botones.

En nuestra aplicación, todo botón de la barra de herramientas se corresponde con algún ítem de un menú. Así pues, aparecen cuatro problemas de mantenimiento de coherencia:

- 1.- Entre el estado del ítem y el del botón: o bien ambos están habilitados o bien están ambos inhabilitados.
- 2.- Entre los nombres del ítem y del botón, así como del texto que aparece en la barra de estado: estos tres nombres son idénticos.
- 3.- Entre los iconos del ítem y del botón: deben ser idénticos.
- 4.- Entre las acciones a realizar al seleccionar el elemento: el resultado será el mismo tanto si seleccionamos el botón como si seleccionamos el ítem.

Para asegurarnos de estas tres características deseables se ha creado la clase `clItemBoton`, que reúne en un solo elemento lógico los dos elementos visuales de la interfaz a la vez que se encarga de mantener la coherencia deseada.

Así pues, para añadir un nuevo botón a la barra de botones, lo que haremos será declarar el ítem asociado como `clItemBoton` en lugar de cómo un `JMenuItem`, ítem simple de Java, y proceder como en el caso anterior.

Insertar una nueva herramienta

Las herramientas de nuestra aplicación, se han desarrollado como Java Plug-Ins, esto es, clases que directamente se *acoplan* a cualquier aplicación gracias a que se autogestionan. Por lo tanto, para agregar una nueva herramienta, al menú y a la barra de herramientas, lo que debemos hacer es agregar un `clHerramienta`, análogo a `clItemBoton`, pero utilizando los datos del Plug-In para decidir la acción a realizar al seleccionar los elementos, al menú de herramientas como si fuera un ítem corriente.

16.7 ANEXO G:

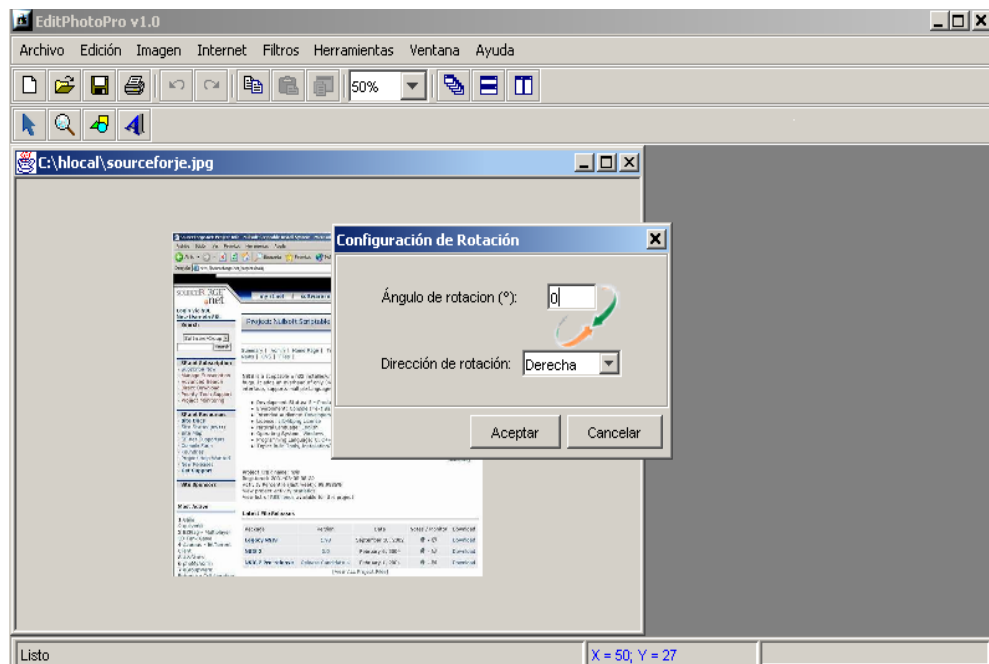
Cómo añadir una nueva transformación

Entendemos por transformación cualquier operación que se pueda realizar sobre una imagen, ya sea el paso de un filtro, la detección de bordes, su binarización, etc.

Las transformaciones serán fundamentales a la hora de añadir nuevas funcionalidades a la aplicación. Por ello hemos intentado que su forma de uso fuese lo más sencilla y abstracta posible, para así facilitar su uso y el futuro trabajo que se pueda realizar sobre la aplicación.

Para añadir una nueva transformación a la aplicación hay q seguir estos sencillos pasos:

1. *Crear una clase que implemente la clase base de las transformaciones (EPP.filtros.clFiltroBase).* Para ello simplemente deberemos implementar la constructora y los métodos: `m_pA_b_cofigurarFiltro` y `m_p_v_aplicarFiltro`. El primero de dichos métodos configura la transformación, mediante la petición de datos al usuario usando un formulario, por ejemplo, y el segundo contiene el código en si de la transformación. Esta nueva clase deberá alojarse en el paquete `EPP.filtros`
2. *Crear un formulario para configurar el filtro, si es necesario.* El formulario deberá formar parte del paquete `EPP.interfaz`. Dicho formulario deberá tomar como base la clase `clFormularioBase` (declararse hijo de ella), lo cual nos facilitará en gran medida la creación del nuevo formulario. En la siguiente figura se puede ver un ejemplo de la utilización de un formulario para configurar la transformación.



Con estos dos pasos ya tendríamos creada la transformación. Ahora quedaría integrarla en la aplicación, lo cual, debido a la abstracción utilizada es muy sencillo y transparente para la aplicación. Es decir, la aplicación trata cualquier transformación del mismo modo, no necesita saber que transformación va a aplicar. A continuación detallamos como integrar el filtro en la aplicación.

3. *Crear una nueva opción de menú desde donde poder invocar la aplicación de una transformación. Ver Anexo F, “Cómo añadir elementos a la interfaz”*
4. *Al activarse dicha opción de menú debemos aplicar la transformación que deseamos se realice. Para ello en primer lugar creamos una instancia de la clase que implemente la transformación que deseamos realizar, que por supuesto, será una clase hija de la clase EPP.filtros.clFiltroBase. A continuación realizamos una llamada al método m_v_aplicarFiltro de la clase clUtilidades (alojada en el paquete EPP.utilidades), pasándole como argumento dicha clase.*

Mediante esta llamada la aplicación se ocupa de aplicar la transformación. En ningún paso siguiente la aplicación sabrá que transformación está aplicando. Sólo lo sabe en el paso 4.

Como se ha podido ver añadir una nueva transformación es muy sencillo. Esto, creemos, facilitará la inserción de nuevas funcionalidades en la aplicación, lo cual simplificará en gran medida el trabajo futuro que otro equipo pueda realizar sobre la aplicación.

16.8 ANEXO H: **Instalación**

16.8.1 CÓMO EJECUTAR LA APLICACIÓN

Para ejecutar Edit Photo Pro v1.0, es necesario copiar el contenido de la carpeta EditPhotoPro en la ruta destino elegida por el usuario para albergar la aplicación. A continuación sólo es necesario hacer doble click sobre el archivo epp.bat.

16.9 ANEXO I: **Manual de Usuario**

APÉNDICE

Edit Photo Pro v1.0 es una aplicación para el *Tratamiento de Imágenes Digitales* y su posterior publicación en *Internet* desarrollada por alumnos de la Facultad de Informática de la Universidad Complutense de Madrid.

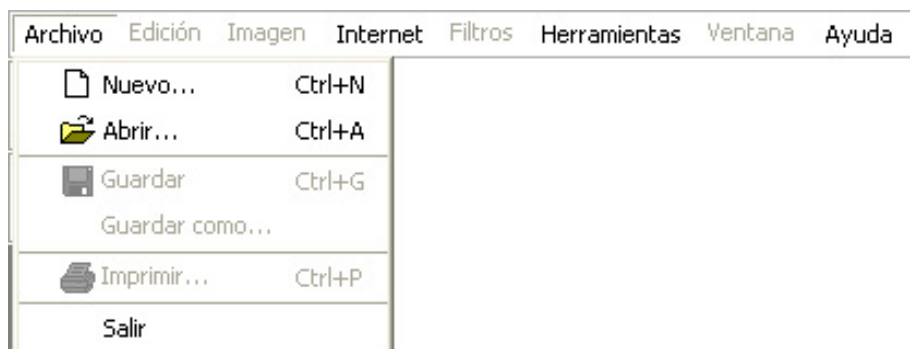
Esta aplicación pretende dar respuesta a la demanda de los usuarios que desean utilizar un mismo programa para realizar las operaciones más comunes y sencillas en el proceso de postproducción fotográfica.

Edit Photo Pro v1.0 nos permite abrir nuestras imágenes digitales, retocarlas, guárdalas agrupadas en carpetas y generar un álbum que si disponemos de conexión a Internet, podremos disfrutar on-line automáticamente.

Para llevar a cabo estas operaciones, presentamos a continuación una descripción de los diferentes menús de la aplicación:

1. El menú Archivo

El menú de archivo nos permite efectuar las operaciones básicas de cualquier programa de tratamiento de imágenes:



Nuevo


☞ Si se pincha en la opción nuevo se abre un archivo con una imagen vacía en la que se puede elegir el ancho y el largo de la misma. Puede elegir el color del fondo de la imagen.

Abrir

📁 Si se pincha Abrir la aplicación presenta un cuadro de diálogo donde se puede elegir el archivo que se desee. Este cuadro de diálogo incorpora en el lado derecho un preview de la imagen. Se pueden abrir simultáneamente varias

imágenes si son seleccionadas. Al pulsar Aceptar a imagen es abierta. En la figura mostramos el cuadro de diálogo.


Guardar

 Al pulsar la opción Guardar pueden ocurrir dos cosas. Si el archivo había sido previamente guardado o existe en la memoria física (disco duro) un archivo con el contenido de la imagen en cuestión, se sobrescribe el archivo de imagen abierto. Si por el contrario se pulsa guardar y no existe ningún archivo desde el cual se haya recuperado la imagen a guardar se abre la opción de Guardar como.

Guardar como

Si se pulsa la opción Guardar como, se abre un cuadro de diálogo muy similar al abierto en la opción abrir. Se puede elegir un archivo ya existente o introducir un nombre para el archivo y especificar, con el filtro disponible en el cuadro la extensión del archivo que queremos crear.

Imprimir

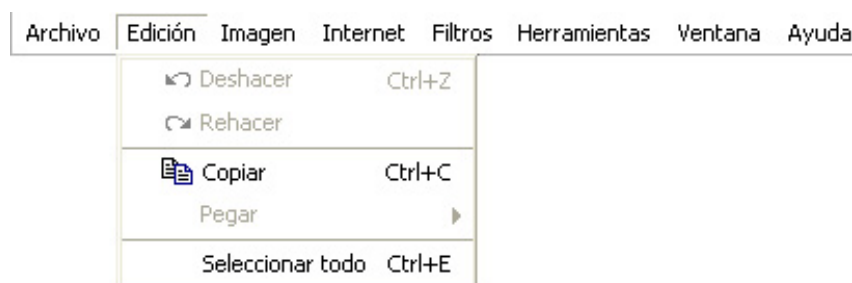
 Al pulsar la opción Imprimir se hace una llamada al cuadro de impresión del sistema y se puede realizar una impresión adecuada de la imagen seleccionada.

Salir


La opción salir cierra la aplicación. Se cierran antes todas las ventanas de las imágenes. La aplicación preguntará, en caso de encontrar alguna imagen que ha sido modificada, si se quieren salvar los cambios accediéndose en este caso a la opción de menú de Guardar o Guardar como según sea el caso.

2. El menú Edición.


El siguiente menú que encontramos en la barra de menús es el de Edición que consta de las siguientes opciones.




Deshacer

 Si se pincha en la opción deshacer el programa elimina el último cambio realizado en la imagen dejando a ésta exactamente igual de cómo estaba antes de haber realizado dicho cambio.


Rehacer

 Esta opción produce el efecto contrario a la anterior. Al pinchar en Rehacer lo que se consigue es provocar en la imagen una transformación que se había realizado anteriormente pero que se había eliminado pulsando Deshacer.

Copiar

 Si se pulsa copiar la aplicación almacena en el portapapeles la imagen activa, que luego se podrá recuperar con la opción pegar.

Pegar

 La opción pegar permite recuperar la imagen anteriormente copiada en el portapapeles y poderla pegar de dos formas diferentes en la aplicación. Una de las formas genera un nuevo archivo con la imagen y la otra nos pega la imagen del portapapeles como una nueva capa en la imagen que se tiene activa en la aplicación.

Seleccionar todo

Por el momento esta funcionalidad no tiene implementación...Se ha mantenido como posibilidad para el trabajo futuro con capas.

3. El menú Imagen

En el menú Imagen podemos encontrar las operaciones básicas que se pueden realizar sobre una imagen.



Invertir

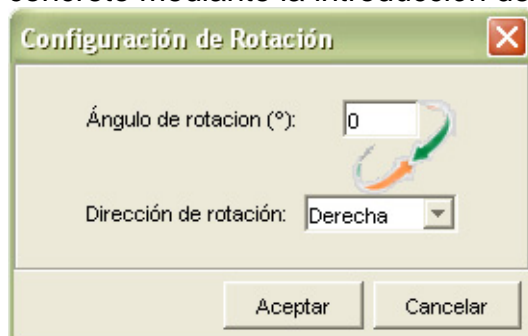
Si se pulsa en Invertir lo que se consigue es la inversión de cada uno de los píxeles de la imagen con su color opuesto (por ejemplo, los píxeles negros pasan a ser blancos y viceversa).

Reflejar

Al pulsarlo se puede elegir entre vertical u horizontal. El efecto que se consigue es una reflexión sobre una línea imaginaria que atraviesa la imagen horizontal o verticalmente según el usuario elija.

Rotar

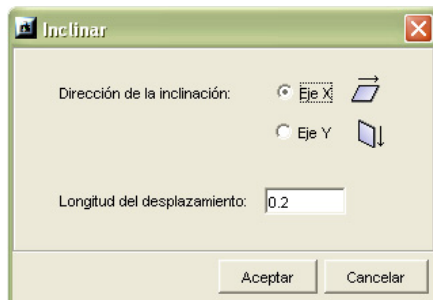
La opción rotar permite girar una imagen un cierto ángulo determinado. Se podrá seleccionar un ángulo predefinido (45°, 90°, etc.) o bien elegir el ángulo concreto mediante la introducción del mismo en el siguiente formulario.



Se introduce el ángulo en el formulario, se elige la dirección hacia la que rotarlo y se pulsa Aceptar para que automáticamente la imagen quede rotada.

Inclinar...

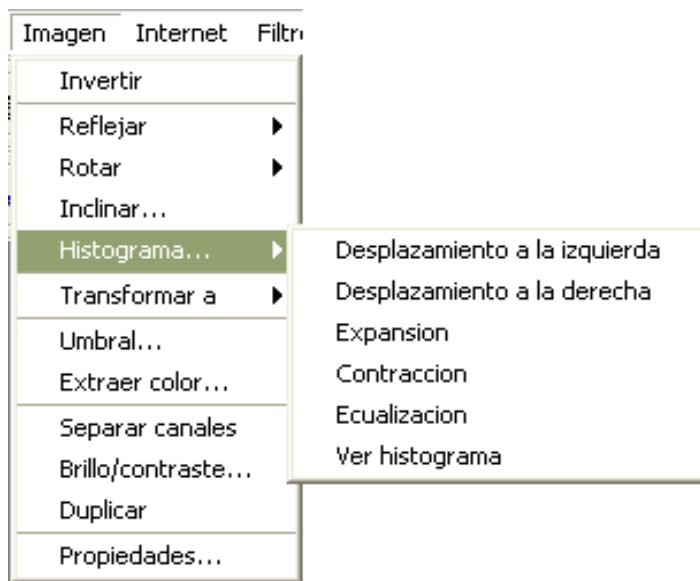
Para inclinar una imagen se pincha esta opción que nos muestra el siguiente formulario.



En este formulario se puede seleccionar si la inclinación será con respecto al eje X o con respecto al eje Y. Además podemos introducir la longitud de desplazamiento (por defecto 0.2).

Histograma...

El histograma de una imagen es una gráfica que contrasta el color de los píxeles con la cantidad de los mismos, es decir se consigue tener una visión de la cantidad de píxeles que tiene cada determinado color de la paleta RGB.



Desde este menú, podremos ejecutar operaciones sobre el histograma de la imagen para cambiar el aspecto de esta.

- Desplazamiento a la izquierda: Como su nombre indica realiza un desplazamiento de todos los píxeles del histograma hacia la izquierda.
- Desplazamiento hacia la derecha: Análogo al anterior pero desplazando los píxeles a la derecha.

- Expansión: El resultado es que el histograma se expande hacia los niveles máximos permitidos
- Contracción: El histograma concentra los píxeles en las regiones centrales.
- Ecualización: Aplicando el algoritmo apropiado produce un realzado de la imagen.
- Ver histograma: Nos permite ver el histograma actual de la imagen.

Para todas estas operaciones de modificación del histograma los tipos de imágenes que contemplan son imágenes a color RGB e imágenes de grises ya que en el caso de las imágenes binarias no tienen sentido los cambios de histograma puesto que el histograma es una línea para el blanco y para el negro.

Transformar a

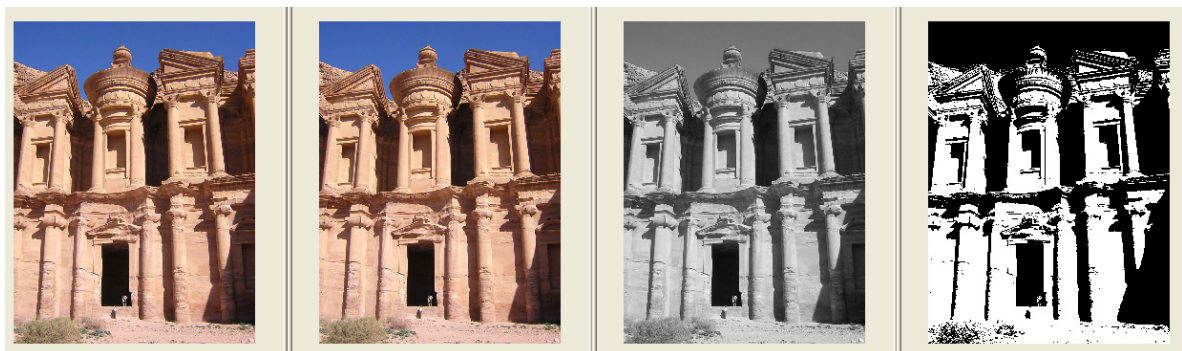


La opción 'transformar a' permite cambiar la paleta de colores a una de las siguientes: IHS, RGB, Grises y Binaria. Consiste en una transformación píxel a píxel de la imagen.

RGB y IHS son dos sistemas de representación de colores. Por tanto no cambia el aspecto de la imagen, solo la representación interna.

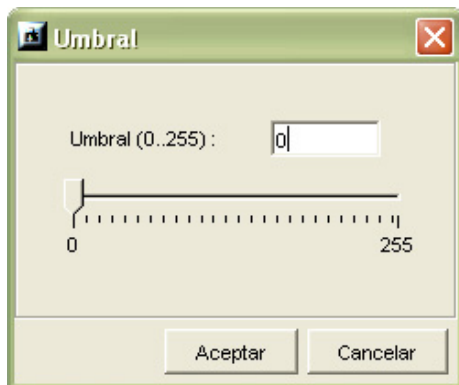
Escala de grises cambia todos los píxeles a diferentes tonalidades de grises.

Binarizar cambia cada píxel a blanco o a negro.



Umbral

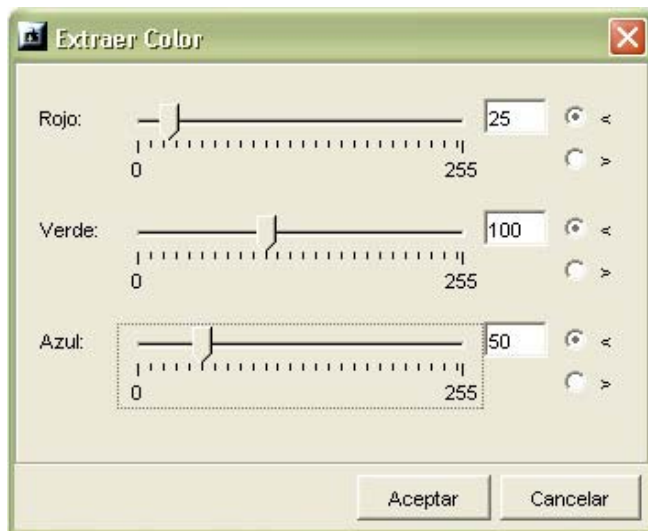
Si se pincha en umbral nos aparecerá un formulario como el que sigue:



En la barra deslizable se elige el parámetro que se quiere aplicar a la transformación. El rango para este parámetro está entre 0 y 255. Una imagen tendrá por defecto 0. Cuanto más próximos al cero nos encontremos más blanco tendrá la imagen. Cuanto más nos aproximemos al 255 más negro tendrá la imagen.

Extraer color

Mediante la opción extraer se consigue modificar los valores de las paletas RGB de la imagen. Con el siguiente formulario se configura la transformación:



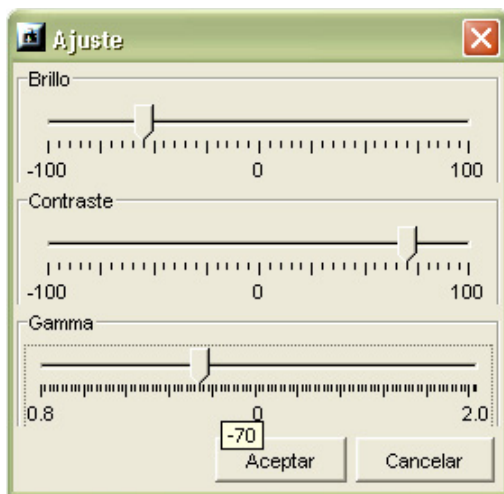
Para introducir un determinado valor se deslizan las barras y se elige si ese valor se quiere añadir o quitar seleccionando los símbolos '<' o '>' respectivamente.

Separar canales

Al pulsar esta opción lo que se consigue es extraer las tres bandas RGB de una imagen e introducir cada una de ellas en una ventana de imagen. La ventana de imagen tendrá el mismo nombre de la ventana padre añadiendo el color de la banda extraída.

Brillo/contraste...

Con la opción Brillo y contraste lo que se consigue es modificar los valores de brillo, contraste y factor Gamma de una imagen. Tendremos para configurar esta transformación el siguiente formulario:



Con las barras deslizantes elegiremos los valores para cada ajuste posible. Al pulsar Aceptar y la transformación se realiza en la ventana activa.

Duplicar...

Esta opción consigue realizar una copia exacta de la imagen que posee la ventana activa e introducirla en otra ventana de imagen.

Propiedades...

Por último, la opción propiedades nos muestra una ventana con las medidas de la imagen y su profundidad de color como se observa a continuación:



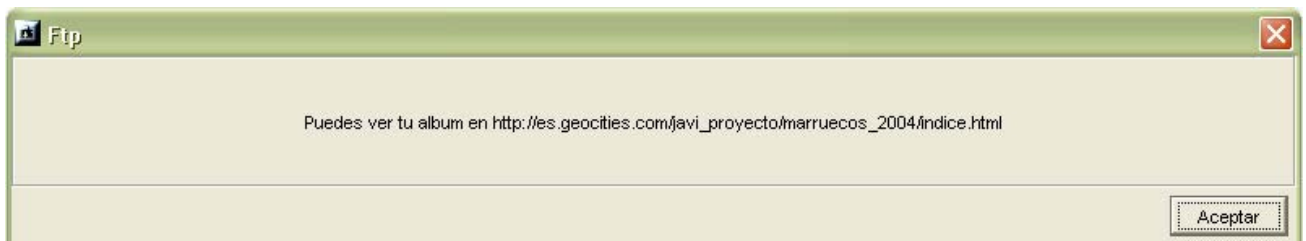
4. El menú Internet

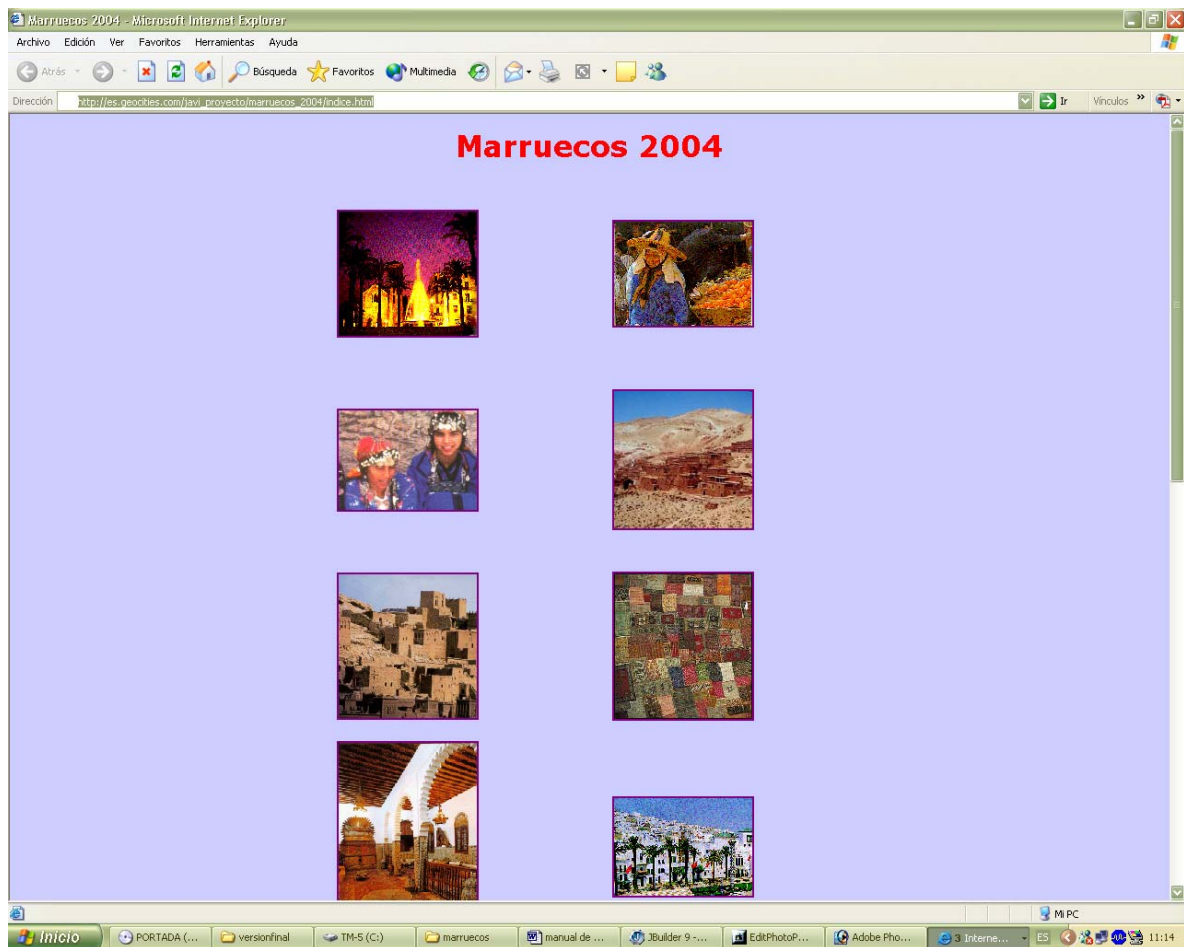
En este menú encontramos las opciones para crear álbumes y publicarlos en Internet:

- Crear Álbum On-line: Crea a partir de los parámetros definidos por el usuario por medio de un cuadro de dialogo (ver figura) una pagina HTML. Posteriormente la sube por ftp al servidor de contenidos Web.



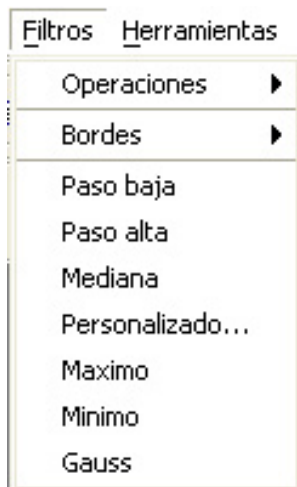
El usuario define los parámetros para crear el álbum HTML. Finalizada la transmisión de ficheros al servidor, el usuario puede visitar su álbum en Internet.
La dirección de la pagina Web aparecerá en un cuadro de dialogo para que el usuario pueda visitarla.





- Configurar ftp: Acción deshabilitada en esta versión. Permitirá al usuario usar una conexión y servidor diferente al provisto por la aplicación.

5. El menú Filtros



Otro de los menús que la aplicación incorpora es el menú de filtros. Los filtros son, en definitiva transformaciones a nivel de píxeles de la imagen. Utilizando un modificador en forma de matriz para los píxeles se logran efectos de suavizado, de perfilado y similares. Cada uno de los filtros definidos aplicará por tanto una transformación a la imagen. Los filtros disponibles son los siguientes:

-Operaciones:

And
Or
Xor
Suma
Resta
Multiplica
Divide

Estas operaciones, no son un filtro propiamente. Son operaciones para combinar imágenes por diferentes métodos. Su aplicación puede ser utilizada para efectos visuales similares a los conseguidos con diversos filtros. La creatividad del usuario dará lugar a múltiples imágenes diferentes por medio de estas operaciones.

-Bordes:

Sobel
Prewitt
Laplaciano
Kirsch...
Frei-Chen

Los bordes nos permiten transformaciones de la imagen en las que se obtienen los bordes de las diferentes superficies de la imagen. Con estas operaciones obtenemos interesantes efectos visuales.



- Paso baja: Suaviza la imagen.
- Paso Alta: Perfila los bordes. Da un efecto de mayor enfoque.
- Mediana, Personalizado, Máximo, Mínimo y Gauss, son filtros de distorsión de la imagen.

6. El menú Herramientas

La aplicación ofrece la posibilidad de modificar imágenes utilizando ciertas herramientas. En el menú de herramientas se puede elegir cada una de estas herramientas. La aplicación también incorpora una barra de herramientas desde la cual aplicar las mismas. Para cada una de las herramientas y desde el menú ver se puede obtener una ventana para configurar la herramienta (si es que dispone de ella). Las herramientas que se pueden utilizar son:

Selección



Selecciona una determinada región de la imagen. Después de pulsar sobre esta herramienta se pasa a la ventana activa y se pincha con el ratón. Se mantiene pulsado el ratón para delimitar una zona con forma rectangular en la imagen.

Observación: Esta herramienta se puede utilizar para reencuadrar imágenes. Selecciona el área que deseas reencuadrar y luego pégala (Ver menú Edición, Pegar) como un nuevo documento.

Lupa



Al pinchar en la opción lupa aparece en la ventana activa una lupa que aumenta la zona de la imagen por donde pasa. Se puede arrastrar esa lupa a través de toda la imagen.

Forma



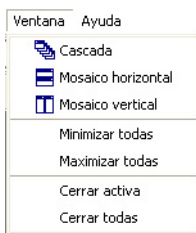
Esta herramienta permite dibujar líneas sobre la imagen para delimitar zonas, crear superficies ralladas, etc....

Texto






También en el menú de herramientas se tiene la opción de introducir texto en la imagen.

7. El menú Ventana



El menú ventana es el menú que permite la colocación de las ventanas de imagen dentro de la aplicación así como operaciones típicas de una ventana como son maximizar, minimizar o cerrar.

En cuanto a las posibles ordenaciones de las ventanas que se pueden elegir son: cascada , mosaico horizontal  y mosaico vertical . Y en cuanto a operaciones típicas de una ventana se tiene: maximizar todas, minimizar todas, cerrar activa y cerrar todas.

8. El menú Ayuda

Desde este menú podemos ver una ventana con el nombre de la aplicación, versión y creadores.



Los abajo firmantes, M^a Luisa Carles Gil, Javier Espejo y Cristina García Aguirre, alumnos de la asignatura Sistemas Informáticos y creadores del proyecto “*Diseño de un interfaz gráfico para procesamiento avanzado de imágenes*”, autorizamos a la Universidad Complutense a difundir y utilizar con fines académicos y no comerciales el mencionado proyecto, tanto la memoria como el código, la documentación y el prototipo desarrollado.

Madrid a 30 de Junio de 2004